Research On Enterprise Production Decision Based on Monte Carlo Simulation and Dynamic Programming Algorithm

Zhuochao Huang [#], Mengxi Yang [#], Wei Zhao [*]

School of mathematics and statistics, Guangdong University of Technology, Guangzhou, China, 510006

* Corresponding Author Email: wzhao@gdut.edu.cn

[#]These authors contributed equally.

Abstract. In the increasingly competitive global market environment, companies face multiple challenges in production decision-making, including resource optimization, cost control, and uncertainty management. Traditional production decision-making methods rely on empirical judgment or simplified models, which struggle to handle the dynamism and randomness in complex production processes. Existing optimization algorithms also have limitations such as overly simplified models, high computational complexity, and inadequate handling of uncertain factors. To address these issues, this paper proposes an integrated decision framework that combines multi-stage dynamic programming, Monte Carlo simulation, and greedy algorithms. By dividing production stages through dynamic programming and establishing a global optimization model, Monte Carlo simulation quantifies the impact of random factors, while the greedy algorithm quickly solves local optimal strategies to reduce computational complexity. Experiments show that this method can effectively balance inspection, disassembly, and replacement costs in scenarios involving component assembly and multi-process semi-finished product production. Additionally, this paper reduces resource waste through a pre-interception mechanism for defect rates, enhancing the company's adaptability to market uncertainties. The research provides a decision tool with both efficiency and precision for multi-objective optimization in complex production systems, helping companies achieve dual goals of minimizing costs and maximizing profits in dynamic environments.

Keywords: Monte Carlo simulation, Dynamic Programming, Production Decision.

1. Introduction

In today's highly competitive market environment, companies face the challenge of making complex and dynamic production decisions. In the early days, production decisions were primarily based on experience and intuition, lacking systematic approaches. With the rise of scientific management theory, there was a greater emphasis on standardization and efficiency, which influenced early production planning and control [1]. How to efficiently allocate resources, optimize production processes, and reduce costs to maintain a competitive edge in the market is a critical issue that every company must seriously consider. Production decision-making issues cover the entire process from raw material procurement to product delivery, involving numerous stages and factors, thus requiring the use of scientific methods and tools for analysis and optimization. Algorithms for solving production decision problems continue to emerge, providing diverse decision support for companies.

Currently, research on production decision-making primarily focuses on using various optimization algorithms to improve production efficiency and reduce costs. Some scholars employ dynamic programming methods to address decision-making issues in multi-stage production processes, breaking down complex production processes into multiple stages and achieving overall optimization by finding the optimal decisions at each stage. Rafflesia et al. [2] proposed a decision framework aimed at solving complex production and cost structure problems to ensure profit maximization under different conditions. Other scholars attempt to combine Monte Carlo simulation with dynamic programming to handle uncertainties in production processes. Cruz et al. [3] for instance, integrate mathematical models with Monte Carlo simulation, specifically considering random and non-stationary demand as a tool for planning under uncertain demand conditions, thereby reducing waste rates within the planning scope. Olanrele et al. [4] introduced a new algorithm that combines linear programming and Monte Carlo simulation to manage uncertainties in fast-moving consumer goods production plans and minimize costs. Additionally, some scholars use genetic algorithms to optimize decision-making issues in multi-process production. Wang et al. [5], for example, used genetic algorithms to optimize the assembly process of key components in the production of popular electronic products, solving the problem of calculating sampling inspection quantities.

Despite significant progress in the study of production decision-making problems, existing research methods also have some limitations. First, many models are overly simplified and fail to accurately capture the complexity of actual production processes. Second, some algorithms have high computational costs, making them unsuitable for large-scale production decision problems. Additionally, there is insufficient consideration of various uncertainties in the production process, such as market demand fluctuations and equipment failures. The application of analytical optimization methods in production and inventory management is limited, primarily because simple functions (usually linear or quadratic) are used to model real systems in order to obtain optimal solutions [6-10].

To overcome the aforementioned issues, this paper proposes an enterprise production decision-making method based on multi-stage dynamic programming and optimization algorithms. The method first uses hypothesis testing to evaluate the quality status during production, then combines a minimum cost flow model to optimize the production process and reduce costs. At the same time, Monte Carlo simulation is employed to model various uncertainties in the production process, and dynamic programming and greedy algorithms are used to formulate optimal production decisions. By integrating these methods, this paper aims to provide enterprises with a more comprehensive, efficient, and practical production decision support tool, helping them gain a greater advantage in fierce market competition.

2. Methods

2.1. The basic function of the Monte Carlo simulation

Monte Carlo simulation is a numerical calculation method based on probability and statistical theory and random sampling technology. Its core idea is to generate a large number of independent and identically distributed random samples, and use the law of large numbers (Law of Large Numbers) and central limit theorem (Central Limit Theorem) to approximate the statistical characteristics of complex systems.

The standardization process for Monte Carlo simulation includes the following steps:

(1) The research objectives are defined and mathematical models are established to describe the uncertain characteristics of the system.

(2) Determine the probability distribution (such as normal distribution, uniform distribution) and parameters of the input variables to ensure that the generated random numbers conform to physical or statistical laws.

(3) Random number generation and sampling: The sample is generated by a pseudo-random number generator, and the sampling of complex distributions is realized by inverse transform method (Inverse Transform), acceptance-rejection method (Accept-Reject) or Markov chain Monte Carlo (MCMC).

(4) System simulation and data acquisition: input the random sample into the model for parallel simulation, and record the output results

(5) Statistical analysis and error evaluation: calculate mean, variance, quantile and other statistics, and quantify the estimation error through confidence interval or standard deviation.

2.2. Introduction to greedy algorithm

Greedy algorithm is a heuristic algorithm based on local optimal choice. It selects the optimal decision in the current state (greedy strategy) step by step to approach the global optimal solution. Its characteristics are no backtracking mechanism and depends on the characteristics of the problem to ensure the global optimality. The algorithm process is shown in Figure 1.

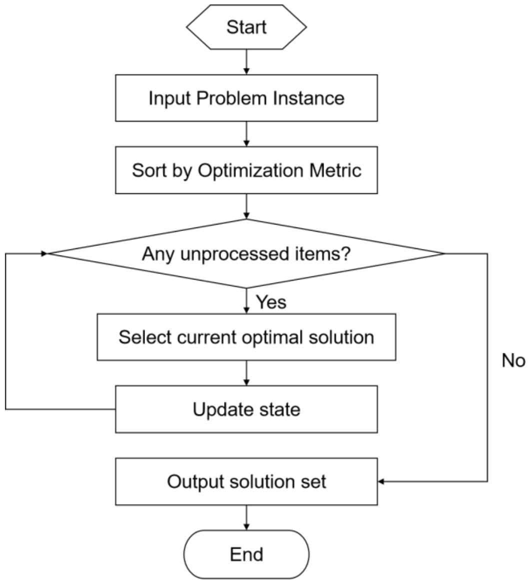

Figure 1. Greedy Algorithm flowchart

The core principle of greedy algorithm

The effectiveness of greedy algorithms depends on two key properties:

Greedy choice property (Greedy Choice Property)

The local optimal choice of each step must be able to derive the global optimal solution, and the choice cannot be traced back. For example, in the activity selection problem, the earliest end of the activity is preferred to reserve more time resources for subsequent steps.

Optimal substructure (Optimal Substructure)

The optimal solution of the problem contains the optimal solution of the subproblem. For example, in the backpack problem, if the current item selection is optimal, then the subproblem selection under the remaining capacity should also constitute the optimal solution.

3. Results

3.1. The situation of direct assembly of finished products with spare parts

Assuming a company produces a popular electronic product that requires the purchase of two types of components (Component 1 and Component 2), and assembles them into a finished product. In the assembled finished product, if just one component is defective, the entire product will be defective; even if both components are qualified, the assembled product may still be defective. For defective products, the company can choose to scrap them or disassemble them. The disassembly process will not damage the components but will incur disassembly costs.

Before assembling the components, the company needs to use sampling inspection methods to decide whether to accept the batch of components purchased from suppliers. Therefore, determining the sample size for the sampling inspection directly affects the efficiency and cost of subsequent production. Assuming that the component pass rate roughly follows a normal distribution, the sample size can be calculated using the normal distribution formula:

$$n = \frac{Z^2 \cdot p(1-p)}{E^2} \quad (1)$$

Where Z is the critical value under the standard normal distribution; p is the nominal defect rate of the parts; E is the allowable error.

Figure 2 shows how the defect rate varies in the confidence interval of 90% and 95% reliability for different sample sizes.

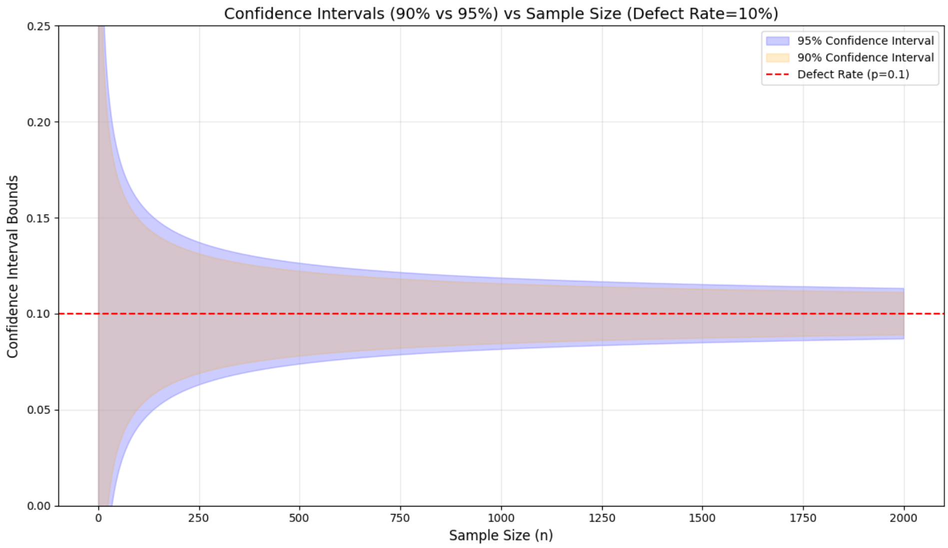

Figure 2. Confidence Intervals VS Sample Size

As the sample size increases, the width of the confidence interval (i.e., the uncertainty of the estimate) gradually decreases. This is because larger samples lead to higher precision in estimation, resulting in a narrower confidence interval. A 95% confidence interval is wider than a 90% confidence interval, indicating that at a higher confidence level, we need a larger sample size to achieve the same confidence interval width.

Now assume that we know six situations of the defect rate of two kinds of spare parts and finished products as shown in Table 1. We are required to make decisions for each stage of the production process:

Table 1. Situations of parts and finished products

Circumstances	Component 1			Component 2			End Product				Defective Product	
	DP	P	TC	DP	P	TC	DP	AC	TC	MP	RL	DC
1	10%	4	2	10%	18	3	10%	6	3	56	6	5
2	20%	4	2	20%	18	3	20%	6	3	56	6	5
3	10%	4	2	10%	18	3	10%	6	3	56	30	5
4	20%	4	1	20%	18	1	20%	6	2	56	30	5
5	10%	4	8	20%	18	1	10%	6	2	56	10	5
6	5%	4	2	5%	18	3	5%	6	3	56	10	40

Through dynamic programming algorithms, the production process of finished goods is broken down into three stages: component inspection for qualification, component qualification testing, and assembly completion, followed by finished product inspection for qualification and disassembly. Decisions at each stage are optimized recursively to gradually achieve the goal of minimizing costs and maximizing profits. By combining the optimal solutions from each stage, companies can make the most advantageous decisions, ensuring production efficiency while minimizing costs and maximizing profits.

Based on the above problems and assumptions, the following mathematical model is established:

$$\min TC = A + B + C + \omega_i \tag{2}$$

$$A = \sum_{i=1}^{2} a_i W_i \tag{3}$$

$$B = B_1 + B_2 + B_3 \tag{4}$$

$$B_1 = b_1 \cdot W_3 \tag{5}$$

$$B_2 = \prod_{i=1}^{2} \left\{1 - \left[1 - p_i \cdot \left(1 - W_i\right)\right]\right\} \cdot b_2 \tag{6}$$

$$B_3 = b_3 \cdot W_4 \tag{7}$$

$$C = \left(1 - W_3\right) \cdot p_3 \cdot c_1 \tag{8}$$

$$W_i = 0 \; or \; 1 \tag{9}$$

$$0 \le p_i \le 1, \omega > 0 \tag{10}$$

The objective function (2) represents the minimum total cost; (3) represents the inspection cost of the spare parts in the inspection stage; (4) ~ (7) represents the inspection, assembly and disassembly costs of the finished products; (8) represents the replacement cost of unqualified finished products; (9) represents the decision state: 1 means execution, 0 means no execution.

The above dynamic programming algorithm mainly focuses on the minimization of total cost. This paper combines the minimum cost flow model to maximize the profit at the same time, and optimizes the production flow in each stage to minimize the cost under the premise of meeting the production demand, so as to maximize the profit in the production process. The specific flow network is defined in Table 2.

Table 2. Composition and interpretation of minimum cost flow

Form	Paraphrase
Edge	Edge has unit flow and cost
Sink Nodes	End point where finished product is sold and qualified
Panel Point	Every network point includes parts inspection, assembly, etc.
Originating Node	Starting point for production, purchase of parts, determines initial quantity.
Traffic Demand	Node costs and required samples determined.

At the same time, this paper randomly generates different situations based on Monte Carlo simulation to simulate whether different numbers of detection, disassembly and replacement are carried out at different stages of the product, so as to estimate the impact of different decision-making methods on the cost and profit of the enterprise.

The solution results of the specific model are shown in Table 3.

Table 3. The solution results of the specific model

Circumstances	1Detection	2Detection	Inspection	Disassembly	Cost	Income	Profit
1	0	0	1	1	2657	4409	1752
2	0	0	1	1	3044	4936	1892
3	0	0	1	1	2657	4409	1752
4	0	0	1	1	2899	4707	1808
5	0	0	1	1	2722	4324	1602
6	0	0	1	0	2982	4788	1806

3.2. Increase the semi-finished product assembly process

In real production processes, multiple procedures and various components are often involved. Suppose an additional step is added between the assembly of components into semi-finished products and the final assembly into finished products. In such cases, making decisions for each stage of the production process becomes more complex. For example, if certain conditions apply to components and semi-finished products as shown in Table 4, and the specific situation of the finished product is illustrated in Table 5.

Table 4. The situation of components and semi-finished products

<table>
<tr><th>Component</th><th>DP</th><th>P</th><th>TC</th><th>Semi-manufactures</th><th>DP</th><th>AC</th><th>TC</th><th>DC</th></tr>
<tr><td>1</td><td>10%</td><td>2</td><td>1</td><td rowspan="3">1</td><td rowspan="3">10%</td><td rowspan="3">8</td><td rowspan="3">4</td><td rowspan="3">6</td></tr>
<tr><td>2</td><td>10%</td><td>8</td><td>1</td></tr>
<tr><td>3</td><td>10%</td><td>12</td><td>2</td></tr>
<tr><td>4</td><td>10%</td><td>2</td><td>1</td><td rowspan="3">2</td><td rowspan="3">10%</td><td rowspan="3">8</td><td rowspan="3">4</td><td rowspan="3">6</td></tr>
<tr><td>5</td><td>10%</td><td>8</td><td>1</td></tr>
<tr><td>6</td><td>10%</td><td>12</td><td>2</td></tr>
<tr><td>7</td><td>10%</td><td>8</td><td>1</td><td rowspan="2">3</td><td rowspan="2">10%</td><td rowspan="2">8</td><td rowspan="2">4</td><td rowspan="2">6</td></tr>
<tr><td>8</td><td>10%</td><td>12</td><td>2</td></tr>
</table>

Table 5. The situation of the finished product

DP	AC	TC	DC	MP	RL
10%	8	6	10	200	40

For the scenario involving multiple processes and numerous components, this paper employs a multi-stage optimization model combined with a greedy algorithm to solve for optimal decisions in local areas. The locally optimal decisions are then integrated with dynamic programming to derive the overall optimal decision for the entire process. In response to specific circumstances, production is divided into three stages: whether components are inspected and semi-finished products are assembled, whether semi-finished products are inspected and finished products are assembled, and whether finished products are inspected and defective finished products that enter the market are dismantled.

Based on the above specific problems and assumptions, the following model is established:

$$\min TC = A + B + \mathrm{C} + D - \omega_i \tag{11}$$

$$A = \sum_{n=1}^{N} D_i \cdot d_i + \sum_{n=1}^{M} E_i \cdot e_i + F \cdot f \tag{12}$$

$$B = \sum_{n=1}^{M} c_i \cdot (1 - p) \tag{13}$$

$$C = \sum c_i \tag{14}$$

(12) represents the total cost of detection, (13) represents the total cost of assembly, and (14) represents the total cost of disassembly, which is a 0-1 variable, where 0 means not to execute and 1 means to execute. The results obtained by combining Monte Carlo simulation and greedy algorithm are shown in Table 6.

Table 6. The results

<table>
<tr><td>C1 Detection</td><td>C2 Detection</td><td>C3 Detection</td><td>C4 Detection</td><td>Cost</td><td>Income</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td rowspan="7">6869.4</td><td rowspan="7">17997</td></tr>
<tr><td>C5 Detection</td><td>C6 Detection</td><td>C7 Detection</td><td>C8 Detection</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>SM1 Inspection</td><td>SM2 Inspection</td><td>SM3 Inspection</td><td>SM1 Disassembly</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>SM2 Disassembly</td><td>SM3 Disassembly</td><td>P Inspection</td><td>P Disassembly</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>0</td></tr>
</table>

3.3. Result analysis

The experimental results show that the multi-stage optimization model proposed in this paper can effectively balance cost control and profit enhancement objectives in production decisions. In the scenario where components are directly assembled into finished products, dynamic programming significantly reduces resource waste caused by fluctuations in defect rates through phased optimization of inspection, assembly, and disassembly strategies. Meanwhile, Monte Carlo simulation provides risk quantification support for uncertain factors, enhancing the robustness of decision-making. The model reduces unnecessary costs while ensuring product quality by flexibly adjusting inspection rates and disassembly strategies, thus verifying the feasibility of multi-stage collaborative optimization.

For complex scenarios involving multiple processes with semi-finished product assembly, the synergy between greedy algorithms and dynamic programming becomes more prominent. The greedy strategy reduces real-time computational complexity through local quick decisions, while dynamic programming coordinates resource allocation across stages from a global perspective, avoiding overall efficiency losses due to local optimality. However, the model still has room for improvement in high defect rates or extreme uncertainty scenarios. Future enhancements could include introducing adaptive learning mechanisms or improving the modeling accuracy of stochastic processes to further enhance the model's adaptability to complex production chains. Overall, this method provides a systematic solution that balances efficiency and precision for multi-objective and multi-constraint production decision-making problems.

4. Conclusions

This paper comprehensively applies the normal distribution hypothesis, multi-stage dynamic programming, Monte Carlo simulation, and greedy algorithm to reasonably simplify calculations while enhancing analytical efficiency: using the normal distribution to quickly estimate pass rates, combined with dynamic programming to divide into multiple stages for global optimal decision-making; employing Monte Carlo simulation to handle production uncertainties, generating random samples to enhance model adaptability; for complex processes, using the greedy algorithm to rapidly solve local optimal solutions, which not only reduces computational complexity but also reduces corporate costs through preemptive rejection of defective products, achieving a balance between precision and efficiency.

There are the following room for improvement and optimization: choose a test method closer to the actual distribution and optimize the sample size requirements, develop a flexible segmentation method to reduce the complexity of dynamic programming, combine local greedy strategy with global backtracking mechanism to balance efficiency and result credibility, so as to enhance the robustness of the model and decision accuracy.

References

[1] Vasconcelos H, Jörke M, Grunde-McLaughlin M, et al. Explanations can reduce overreliance on ai systems during decision-making[J]. Proceedings of the ACM on Human-Computer Interaction, 2023, 7(CSCW1): 1-38.

[2] Rafflesia U, Widodo F H, Angraini T. Dynamic programming for an optimization of production plan[C]//Journal of Physics: Conference Series. IOP Publishing, 2021, 1731(1): 012032.

[3] Cruz J A, Salles-Neto L L, Schenekemberg C M. 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.

[4] Olanrele O O, Ismaila S O, Adeaga O A, et al. Managing Uncertainty in Production Planning for Fast-Moving Consumer Goods: A Linear Programming and Monte Carlo Simulation Framework[C]//2023 International Conference on Science, Engineering and Business for Sustainable Development Goals (SEB-SDG). IEEE, 2023, 1: 01-08.

[5] Wang Y, Yin X. Research on Optimization of Production Decision Based on Dynamic Planning and Genetic Algorithm[C]//2024 IEEE 2nd International Conference on Electrical, Automation and Computer Engineering (ICEACE). IEEE, 2024: 843-848.

[6] Maitra S. A system-dynamic based simulation and Bayesian optimization for inventory management[J]. arXiv preprint arXiv:2402.10975, 2024.

[7] Maitra S, Mishra V, Kundu S. a Novel Approach with Monte-Carlo Simulation and Hybrid Optimization Approach for Inventory Management with Stochastic Demand[J]. arXiv preprint arXiv:2310.01079, 2023.

[8] Perez H D, Hubbs C D, Li C, et al. Algorithmic approaches to inventory management optimization[J]. Processes, 2021, 9(1): 102.

[9] Farizal F, Gabriel D S, Rachman A, et al. Production scheduling optimization to minimize makespan and the number of machines with mixed integer linear programming[C]//IOP Conference Series: Materials Science and Engineering. IOP Publishing, 2021, 1041(1): 012046.

[10] Shi C J L, Bugtai N T, Billones R K C. Multi-Period Inventory Management Optimization Using Integer Linear Programming: A Case Study on Plywood Distribution[C]//2022 IEEE 14th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM). IEEE, 2022: 1-5.