Journal of Industrial and Management Optimization

JIMO, 22 (2): 1140–1167.
DOI: 10.3934/jimo.2026042
Received: 16 October 2025
Revised: 29 December 2025
Accepted: 06 January 2026
Published: 29 January 2026

https://www.aimspress.com/journal/jimo

Research article

Dynamic pricing strategy design for manufacturing service providers in manufacturing platforms

Wenchong Chen[1,2], Xiaoliao Tang[2], Jiehui Qi[3] and Hongwei Liu[4,*]

[1] Experimental Center of Data Science and Intelligent Decision-Making, Hangzhou Dianzi University, Hangzhou, 310018, China
[2] College of Management, Hangzhou Dianzi University, Hangzhou, 310018, China
[3] Technical Committee, New H3C Technologies Co., Ltd, Hangzhou, 310051, China
[4] College of Management and Economics, Tianjin University, Tianjin, 300192, China

* **Correspondence**: Email: 1016209007@tju.edu.cn.

Abstract: Existing game theory and scheduling optimization-based dynamic pricing methods suffer from complex decision-making processes, failing to adapt to service providers' (SPs) dynamic pricing needs in manufacturing service recommendation scenarios. To address this gap, this study proposes four rule-based dynamic pricing strategies—cost-plus pricing (CPP), service recommendation outcome-driven pricing (SROP), service capacity surplus-driven pricing (SCSP), and market average price-driven pricing (MAPP)—by analyzing SPs' dynamic response mechanisms during service recommendation. Additionally, integrating SPs' adaptive learning behaviors in competitive environments, two reinforcement learning (RL)-driven adaptive dynamic pricing methods were developed. These six strategies were embedded into a multi-agent simulation model abstracted from a real service recommendation system to evaluate their performance across diverse market environments. Results show that: (1) Among the rule-based strategies, SROP exhibits superior competitiveness in most scenarios due to its direct linkage with recommendation outcomes; (2) RL-driven pricing methods do not consistently outperform their rule-based counterparts, indicating that one-sided pursuit of dynamic learning capabilities may not help SPs establish market advantages. This study provides actionable pricing references for SPs in manufacturing service platforms and enriches the theoretical framework of dynamic pricing in service recommendation contexts.

Keywords: platform-based manufacturing; manufacturing service recommendation; dynamic pricing; reinforcement learning; multi-agent simulation
Mathematics Subject Classification: 90B50, 68T42

1. Introduction

The advancement of new-generation information technologies, including cloud computing and the industrial Internet, has enabled the manufacturing industry to transform the vision of "interconnection and sharing of everything" into reality. As early as 2000, the United States established MFG, the world's largest manufacturing platform, to facilitate the efficient matching and transactions of production orders and manufacturing capacity. Subsequently, manufacturing platforms targeting different industrial sectors were developed in succession, including Protolabs, Zemi, Yun Cut, and Jiepei. Financial reports from Protolabs and Zemi for 2023 indicate that their market capitalizations have reached approximately USD 1.01 billion and USD 1.735 billion, respectively. This highlights the substantial market potential and promising development prospects of manufacturing platforms. Although these platforms effectively streamline channels for aligning industrial supply and demand and reduce the configuration costs of manufacturing networks, the continuous expansion of service scale inevitably leads to an increasing number of manufacturing requests and registered service providers. Consequently, efficiently matching the supply and demand of numerous manufacturing services has become a critical challenge in the development of such platforms.

Currently, numerous scholars have designed various approaches to address the problem of supply–demand matching (SDM) in different contexts. Some studies focus on the negotiation processes among stakeholders (i.e., service providers, service demanders, and the platform), developing game-theoretic frameworks for SDM modeling [1,2], or exploring SDM involving multiple service providers and demanders through scheduling-related optimization [3–5]. These studies clarify the complex framework of SDM but also highlight a critical issue: the service pricing problem. Pricing holds significant practical relevance, as it affects both the service purchase behavior of demanders and the ultimate outcomes of SDM [6]. Insufficiently, both game-theoretic and scheduling-related approaches aim at addressing the SDM with restricted service providers and demanders. As the number of manufacturing services published on the platform continues to grow, information overload has become an increasingly serious challenge [7]. It poses a significant challenge for inexperienced demanders to manually select candidate services to meet specific functional requirements [8]. Manufacturing service recommendation has been widely regarded as an effective solution to alleviate information overload and has attracted increasing attention from researchers in recent years [9]. In this context, a variety of recommendation algorithms, including collaborative filtering [10], K-medoids clustering [11], K nearest neighbor [12], and so on, have been proposed. Meanwhile, the pricing strategies adopted by service providers when bidding for jobs within such recommendation systems have also emerged as an important research topic. For instance, Zhu et al. [13] designed three pricing strategies under a game-theoretic framework and evaluated their performance through a simulation using the K nearest neighbor recommendation algorithm. These pricing strategies can help service providers adjust their service prices effectively; however, they have the following three limitations:

(1) The game-theoretic and scheduling-related pricing approaches are excessively complex, making them inapplicable for service providers' participation in platform-based manufacturing. Currently, small- and medium-sized enterprises (SMEs) constitute the primary participants in platform-based manufacturing. Constrained by their scale, game-theoretic and scheduling-related approaches are difficult to promote and apply in practice. Based on a specialized survey we conducted on SMEs (e.g., Jianhua Machinery, Guanming Packaging, Yongqiang Auto Parts, and Banghe Die Casting) using the COSMOPlat platform, we found that these enterprises generally adjust service prices dynamically according to their surplus production capacity or the platform's service recommendations. This practical context necessitates the development of concise, rule-oriented dynamic pricing strategies integrated with the platform's service recommendation process.

(2) The prevailing pricing approaches—predominantly game-theoretic and scheduling-related—overlook the adaptive learning behaviors of service providers within the complex context of platform-based manufacturing. While these methods can accurately model service providers' optimal pricing decisions in complex environments, they fail to capture how providers adaptively adjust prices in response to the platform's service recommendation outcomes. In recent years, the rapid advancement of artificial intelligence has drawn significant attention to reinforcement learning-based approaches, which are model-free and well-suited to addressing service pricing challenges in complex settings. However, there remains a lack of research on applying existing reinforcement learning methods to service pricing in platform-based manufacturing.

(3) There is a lack of simulation models capable of depicting the dynamic service recommendation process in platform-based manufacturing to evaluate the effectiveness of different service pricing strategies. Different pricing strategies should be evaluated within realistic simulation environments to assess their effectiveness. Accordingly, some scholars have developed multi-agent–based simulation models for the service matching process in platform-based manufacturing, thereby verifying the effects of different recommendation algorithms on group evolution outcomes [14,15]. However, a limitation of these simulation models is that they do not incorporate service providers' service pricing as a key variable, making them unable to be directly used to assess the actual implementation effects of various service pricing strategies in platform-based manufacturing.

This paper aims to address the aforementioned limitations by formulating a focused research question: Against the backdrop of an increasingly competitive platform-based manufacturing landscape, how should service providers develop rule-oriented or reinforcement learning-based dynamic service pricing strategies to gain a competitive edge in the market, secure more service recommendations, and ultimately maximize profits? By tackling this research question, this study makes three key contributions, which are elaborated as follows:

(1) Four rule-oriented dynamic pricing strategies for service providers are developed. These strategies comprehensively consider key factors such as service providers' costs, manufacturing platforms' service recommendation results, residual service capacities, and average market service prices, enabling efficient modeling of the differentiated pricing preferences of various service providers in the platform-based manufacturing environment. Compared with the pricing policies presented in references [13] and [15], the pricing strategy proposed in this study takes platform service recommendation volume and corresponding revenue as the core basis for dynamic price adjustment. This mechanism enables SPs to respond more precisely to the platform's service recommendation

process. In addition, we further embed four distinct rule-based pricing strategies into different SP agents; this allows for the precise identification of applicable scenarios for each pricing strategy under competitive conditions, thereby furnishing a decision-making basis for SPs to select optimal pricing strategies in accordance with their own endowment conditions.

(2) Two adaptive dynamic pricing strategies based on the Q-learning algorithm are proposed. One strategy involves learning to optimize the four aforementioned rule-based pricing strategies, while the other focuses on direct price learning to determine optimal pricing decisions. The applicability of the two strategies is evaluated by comparing their performance across diverse market environments. Beyond the common practice adopted in [16–18], where service prices are regarded as the core learning target, this study not only incorporates prices into the learning scope but also identifies dynamic pricing strategies as another key learning parameter. This design not only enriches the pricing learning mechanism of service providers (SPs) but also establishes a brand-new pricing decision-making paradigm for their reference.

(3) A multi-agent simulation model is developed to evaluate the performance of dynamic pricing strategies across various artificial market environments. The model simulates how different pricing approaches compete to secure service recommendations and is designed to assess the adaptability and survivability of these strategies in diverse competitive settings. In contrast to the studies reported in [14], [15], and [19], which only focus on the service recommendation process and the evolution process of the platform-based manufacturing ecosystem, the simulation model constructed in this study further deconstructs the pricing strategies and decision-making behaviors of SPs, and conducts multi-agent modeling analysis. This effectively addresses the deficiency of existing multi-agent simulation models in depicting the behavioral mechanisms of SPs.

This research provides pricing decision support for small- and medium-sized manufacturing enterprises participating in service competition, while also offering an experimental platform for manufacturing service platforms to simulate and predict service recommendation outcomes. The remainder of the paper is organized as follows: Section 2 presents a literature review on pricing behaviors, dynamic pricing strategies, and the use of simulation models for service recommendation in platform-based manufacturing. Section 3 outlines the problem statement and research scope. Section 4 details the modeling of various dynamic pricing strategies. In Section 5, a multi-agent simulation framework is established by encapsulating different entities into an agent-based system. Section 6 describes the experimental design and simulation implementation. Finally, Section 7 summarizes the research findings and discusses their theoretical and practical implications.

2. Literature review

2.1. Pricing behavior and modeling in platform-based manufacturing

Pricing in platform-based manufacturing systems is jointly influenced by consumer structure, bargaining behavior, and demand characteristics. Peng et al. [6] showed that when the proportion of "price takers" in the consumer population is relatively high, and demand is linear, service providers tend to raise their posted prices to increase revenue; however, under elastic demand, as the share of "price takers" increases, providers tend to lower prices to better adapt to market conditions. Bisceglia

et al. [20] conducted an empirical study in highly competitive markets with sluggish demand, examining the dynamic relationship between transaction volume and price volatility. They found that when production cost convexity is either relatively high or low, the optimal pricing strategy involves punitive or incentive pricing for higher output levels. Gallego et al. [21], in their study on single-product pricing, demonstrated that optimal prices decrease as residual service capacity increases and remaining selling time decreases. In research on pricing under different contractual arrangements on shared manufacturing platforms, Wu et al. [22] found that when the platform and manufacturers adopt a revenue-sharing contract, the platform may set higher prices to reflect market expansion resulting from collaboration. Bi et al. [23] examined how different governance structures on networked collaborative manufacturing platforms affect product pricing strategies and reported that when the platform is manufacturer-led, manufacturers exhibit more stable pricing, whereas when the platform is designer-led, higher commission rates incentivize the platform to lower the price of standardized products. Sun et al. [24] investigated optimal pricing strategies for 3D printing platforms that simultaneously offer standardized and customized products. Their findings indicated that the price of standardized products is positively correlated with their own quality and negatively correlated with the quality of customized products; meanwhile, under low labor costs, the price of customized products is positively correlated with their own quality and negatively correlated with the quality of standardized products.

There are complex interactions among service providers, platforms, and demand-side users, and many studies employ game-theoretic models to examine pricing behavior within these interactions. To address how service providers in cloud manufacturing systems should set prices under the joint influence of heterogeneous consumer structures, bargaining behaviors, and different demand forms, Peng et al. [6] proposed a three-party game-based dynamic pricing framework. To mitigate price fluctuations caused by competition for cloud manufacturing service resources, Tao et al. [25] developed a pricing game model grounded in multi-agent competition, revealing the interactive mechanisms underlying pricing strategies among service suppliers. Focusing on the heterogeneity of resource supply in cloud manufacturing environments, Wang [26] introduced a bilateral game model to determine the equilibrium price between service providers and demand-side users. Wu [22] constructed four cooperative advertising models using differential game theory and compared the decision mechanisms under different contractual arrangements, offering optimal cooperation strategies for manufacturers and platforms. To investigate how suppliers' resource-sharing strategies in cloud manufacturing affect customer satisfaction and profit allocation, Cao et al. [27] developed a two-stage model based on Stackelberg games, analyzing the equilibrium pricing and task allocation mechanisms between suppliers and cloud platform operators.

2.2. Dynamic pricing strategies in platform-based manufacturing

As a highly competitive strategy in complex market environments, dynamic pricing has long attracted substantial scholarly attention [28]. Besbes et al. [29] investigated a multi-period single-product pricing problem under an unknown demand curve and found that, under certain market conditions, adopting a linear demand model to adjust prices dynamically across periods can achieve near-optimal revenue performance. Zhu et al. [13] showed that service providers in cloud

manufacturing systems dynamically adjust their pricing strategies based on capacity conditions: when idle, they adopt competitive low pricing by reducing management fee rates to increase the probability of winning bids; when saturated, they pursue premium pricing to maximize profits; and under normal load, they maintain average pricing to ensure reasonable returns. Wu et al. [30] demonstrated that in equipment maintenance scheduling on cloud manufacturing platforms, high time sensitivity and demand uncertainty drive service providers to implement dynamic pricing based on factors such as cost and waiting time, thereby improving response speed and operational efficiency.

From a platform perspective, Zhang et al. [17] proposed a personalized dynamic pricing strategy that incorporates estimates of suppliers' short- and long-term preferences and developed a corresponding dynamic price-driven collaborative optimization method for manufacturing services. Using a Q-learning algorithm, they demonstrated that this integrated approach can significantly enhance system performance and optimization outcomes. Zheng et al. [31] employed a Q-learning algorithm to study multi-period joint pricing and ordering decisions for perishable products. By optimizing decision variables—including new product order quantities, sales prices, and carryover quantities of older inventory—the model achieves profit maximization across periods.

Several studies base price adjustments on predictions of future demand, job arrival rates, or resource utilization intensity. These works employ time-series models, neural networks, or hybrid machine-learning approaches to construct dynamically updated pricing strategies; the methods predict future prices or resource utilization and feed the forecasts into an optimizer, enabling service providers to set prices that enhance revenue [32,33]. Ma et al. [34] integrated demand forecasting with competitive analysis by combining price-elasticity estimation and competitor behavior modeling. Through the coordinated use of optimization algorithms and revenue-management techniques, they developed a dynamic pricing system capable of real-time adjustments, helping firms maximize revenue in markets for perishable goods and services.

2.3. Multi-agent simulation model for platform-based manufacturing

With the growing adoption of cloud manufacturing and platform-based production models, researchers have increasingly employed multi-agent simulation techniques to construct virtual manufacturing ecosystems that replicate the dynamic interactions among users, service providers, and platforms. This approach enables low-cost analysis of service-matching mechanisms and system behaviors. Zhao et al. [35] developed a multi-agent-based cloud manufacturing transaction simulation platform that models enterprises as autonomous service agents, enabling full-process simulation of order release, service discovery, and capability matching. The platform provides a systematic framework for evaluating how matching mechanisms and workflow design affect operational efficiency. Building on this work, Zhao [36] proposed a service-agent network model that incorporates Quality of Service (QoS) attributes, service feedback mechanisms, and self-organizing behaviors, allowing service matching to more accurately reflect real market conditions. The model simultaneously captures the processes of service evaluation updates and dynamic relationship evolution.

To enhance the efficiency of task-resource matching and reduce resource consumption, Hu et al. [37] constructed a multi-layer matching framework centered on resources, tasks, and constraints. By modeling multidimensional attributes—including resource capabilities, time windows, machining

quality, and cost—this framework enables precise alignment between manufacturing resources and user requirements. Zhang et al. [38] further introduced a simulation model for dynamic supply-demand matching in cloud manufacturing, designed to evaluate the effects of matching rules, platform response mechanisms, and resource distributions on overall platform efficiency and ecosystem sustainability. Guo et al. [39] proposed an agent-based service discovery method that, through simulation experiments, demonstrated how search strategies, capability descriptions, and filtering rules influence matching efficiency. Their service-feature modeling framework provides a theoretical foundation for subsequent research on service matching and recommendation. Li et al. [40] developed a distributed multi-agent resource-sharing system for manufacturing, in which resource, scheduling, and request agents interact collaboratively to optimize cross-organizational resource allocation and task distribution, effectively coordinating multi-party supply–demand relationships.

2.4. Research gaps

Existing studies have proposed various pricing methods for manufacturing services in platform-based manufacturing environments. These methods effectively capture the interests of three parties—service providers, service demanders, and manufacturing platforms—along with their interactive decision-making processes. However, a notable limitation is that these methods primarily focus on constructing quantitative decision models. For small- and medium-sized enterprises (SMEs) participating in platform-based manufacturing, such models are not only operationally complex but also struggle to accurately capture the dynamic and evolving nature of the service recommendation process. Reinforcement learning algorithms can support the development of dynamic pricing strategies with autonomous learning capabilities; however, relatively few studies have applied them to manufacturing service pricing to date. Moreover, different pricing strategies should be evaluated within realistic simulation environments to assess their effectiveness. Accordingly, some scholars have developed multi-agent-based simulation models for the service matching process in platform-based manufacturing, thereby verifying the effects of different recommendation algorithms on group evolution outcomes [14,15]. However, a limitation of these simulation models is that they do not incorporate service providers' pricing as a key variable, preventing direct assessment of the actual effects of different service pricing strategies in platform-based manufacturing.

3. Problem description

Platform-based manufacturing integrated with service recommendation involves three types of decision-making entities: service providers (SPs), the service recommendation executor (SRE), and service demanders (SDs). Specifically, SPs encapsulate their manufacturing capabilities as services and upload them to the platform. SDs can conveniently search for and utilize these services via the platform and provide rating feedback upon service completion [41]. The manufacturing platform leverages new-generation information technologies to achieve ubiquitous connectivity of manufacturing services, creating a vast pool of manufacturing service resources. Encapsulated within the platform, the SRE is responsible for recommending the optimal manufacturing service to service demanders upon receiving their service requirements [42].

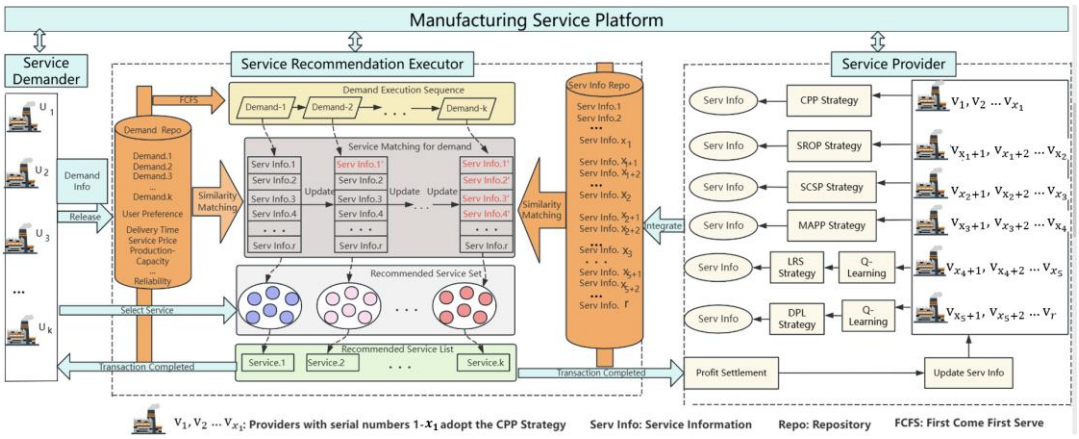


Figure 1. Recommendation processes in platform-based manufacturing.

The recommendation processes in platform-based manufacturing are illustrated in Figure 1. These processes can be described in the following four steps:

Step 1: Manufacturing service demand generation. Manufacturing service demanders arrive at the manufacturing platform randomly, represented by the set $U = \{u_1, u_2 \dots, u_n\}$, where each demander is indexed by $j = 1, \dots, n$. They upload their manufacturing service requirement information to the platform, denoted by the set $D = \{D_1, D_2 \dots, D_n\}$. Here, $D_j = \{d_{j1}, d_{j2} \dots, d_{jm}\}$ represents the service requirements of demander u_j in the platform, where d_{jk} is the expected value of demander u_j for the k-th service attribute ($k = 1, \dots, m$), such as the desired service price, the desired service quality, and other relevant attributes.

Step 2: Manufacturing services filtering and recommendation. We use set $V = \{v_1, v_2 \dots, v_n\}$ to represent manufacturing service providers, indexed by $i = 1, \dots, r$, and the set $B = \{B_1, B_2 \dots, B_r\}$ to represent the service attributes of all providers, where $B_i = \{b_{i1}, b_{i2} \dots, b_{im}\}$ denotes the values of each service attribute of provider v_i. The service recommendation executor calculates the similarity $Sim(D_j, B_i)$ between demander u_j and provider v_i using the Euclidean distance method and performs service recommendation based on this similarity. This recommendation mechanism is simple and mature, making it widely adopted in practical applications [43]. The calculation for $Sim(D_j, B_i)$ is shown in Equation (1):

$$Sim(D_j, B_i) = \sum_{k=1}^{m} \left(\bar{d}_{jk} - \bar{b}_{ik}\right)^2 \tag{1}$$

where $\bar{d}_{jk}$ and $\bar{b}_{ik}$ denote the normalized values of each attribute, which are obtained via the min-max normalization method. We set a recommendation threshold δ and all service providers with $Sim(D_j, B_i) < \delta$ are placed into the candidate service set V_j' which will be recommended to service demander u_j. We sort service demanders in accordance with the first-come-first-served rule and sequentially recommend the set of candidate service providers to them.

Step 3: Service provider selection and capital settlement. The service demander evaluates the services from providers within the candidate set V_j' based on their preferences for the various attributes. The service provider that meets the capacity requirements and has the highest rating is selected. A service contract is then signed with the relevant provider, followed by capital settlement.

Step 4: Service price update. Each SP adjusts its service price according to the predefined rules, which constitute the core strategy this paper endeavors to propose. The process then returns to **Step 1** to initiate the service recommendation and selection processes for the next time period.

We aim to develop a set of simple and operable dynamic pricing strategies to support SPs in their pricing decision-making, with full consideration of the aforementioned service recommendation processes.

4. Dynamic pricing strategy modeling for service providers

This section proposes several dynamic pricing strategies for service providers, integrating the service recommendation processes in platform-based manufacturing. Building on previous research, four rule-based dynamic pricing strategies and two self-adaptive, learning-driven dynamic pricing strategies are developed. The four rule-based strategies include cost-plus pricing (CPP), service recommendation outcome-driven pricing (SROP), service capacity surplus-driven pricing (SCSP), and market average price-driven pricing (MAPP). CPP is typically suitable for scenarios with relatively stable cost structures and a high degree of product standardization. In such contexts, enterprises can accurately calculate costs and subsequently set prices based on expected profit margins [44]. SROP is particularly suitable for highly competitive markets with significant service homogenization, where service providers face intense rivalry for service demander orders [14]. It is also well-suited to scenarios where real-time feedback on recommendation volume and transaction profits is readily available, enabling dynamic price adjustments to balance market share expansion and profit optimization. Additionally, a primary motivation for service providers to participate in platform-based manufacturing is excess capacity. Engagement on such platforms enables full utilization of surplus service capacity, resulting in a capacity surplus-driven pricing strategy [45]. The final rule-based strategy is MAPP [45], which helps providers avoid customer loss due to unduly high prices or profit erosion from unduly low prices.

Q-learning, as an efficient learning method, is well-suited for addressing self-adaptive learning problems with discrete action and state spaces [46]. Drawing on Q-learning principles, we develop two self-adaptive, learning-driven pricing strategies: one involves learning to optimize the four aforementioned rule-based pricing strategies (denoted as LRS), and the other focuses on direct price learning to determine optimal pricing decisions (denoted as DPL). DPL dynamically adjusts service prices based on real-time market conditions, learning to increase or decrease prices in response to the specific context of service recommendations. LRS is well-suited for scenarios with clear pricing rules and stable market logic, offering strong interpretability and low computational complexity, although it may be limited by the original rules. In contrast, DPL is more applicable to complex, volatile markets with ambiguous pricing rules, enabling flexible adaptation to unforeseen market changes. The applicability of the two strategies is evaluated by comparing their performance across diverse market environments.

4.1. Modeling for rule-based dynamic pricing strategies

Given the relative simplicity of the pricing logics underlying CPP and MAPP, instead of providing elaborate details on them, this section focuses on the two rule-based pricing strategies: SROP and SCSP.

4.1.1. Modeling for SROP

Building on the work of Xue et al. [14], we have refined the decision-making conditions: whereas Xue et al. relied solely on a single utility value to trigger price adjustments, our approach incorporates a dual-factor coordination mechanism—integrating service recommendation volume and profit—into SROP. This design not only mitigates potential misjudgments of market supply-demand imbalances arising from reliance on a single metric but also establishes dynamic benchmarks that autonomously adapt to market fluctuations. Specifically, if an SP receives a high volume of service recommendations but achieves relatively low final profit, it will appropriately increase its price. Conversely, if an SP receives a low volume of service recommendations but achieves a relatively high final profit, it will lower its offering price. Otherwise, the SP maintains its current price. This pricing strategy helps SPs balance service recommendation volume and profit attainment. The SROP is formally expressed in Equation (2):

$$p(t+1)\begin{cases} p(t)+\beta*\mathrm{P_{avg}}*10\%, & (TR \geq TR_{avg}) \wedge (CP \leq CP_{avg}) \\ p(t)-\beta*\mathrm{P_{avg}}*10\%, & (TR < TR_{avg}) \wedge (CP > CP_{avg}) \\ p(t) & \text{else} \end{cases} \tag{2}$$

where t is the current evolution period, P_{avg} is the average price of service, TR is the volume of service recommendation, TR_{avg} is the mean volume of service recommendation, CP is the service recommendation profit, and CP_{avg} is the mean profit of service recommendation. The β denotes the risk preference coefficient of the SP.

4.1.2. Modeling for SCSP

SCSP is developed based on Chang et al. [45]. Each SP dynamically adjusts its prices at the end of each evolution period according to its current remaining service capacity. Specifically, if the current remaining capacity exceeds the standard remaining capacity I_{st}, the SP will set a lower price; conversely, the price will be increased. The specific pricing logic is formally defined by Equations (3) and (4).

$$I_{st} = \left(1 - \frac{t}{\tau}\right) * Q \tag{3}$$

$$p(t) = \begin{cases} p_{min} & f_{in} \cdot \left(1 - \dfrac{I_{cu}}{I_{st}}\right) \leq 0 \\ p_{min} + f_{in} \cdot \left(1 - \dfrac{I_{cu}}{I_{st}}\right) & 0 < f_{in} \cdot \left(1 - \dfrac{I_{cu}}{I_{st}}\right) \leq p_{max} - p_{min} \\ p_{max} & f_{in} \cdot \left(1 - \dfrac{I_{cu}}{I_{st}}\right) > p_{max} - p_{min} \end{cases} \tag{4}$$

where τ denotes the service capacity refresh cycle, Q represents the total capacity of the SP, p_{min} and p_{max} denote the lower and upper price bounds, respectively, f_{in} is the influence coefficient of the remaining capacity on pricing, and I_{cu} denotes the current remaining capacity of the SP.

4.2. Modeling for self-adaptive learning-driven dynamic pricing strategies

During initial system evolution, self-learning SPs randomly explore pricing strategies to participate in market competition. Subsequently, these providers update their Q-tables based on revenue obtained from market interactions and optimize their decision-making through the Q-learning algorithm to transition to subsequent states. The state transition process for these self-learning pricing models is illustrated in Figure 2.

4.2.1. State space definition

As the core variable in implementing dynamic pricing strategies, the service price serves as a "bridge" linking an SP's decisions to market responses. Meanwhile, capital value directly reflects the practical impact of pricing decisions on a service provider's survival and development. Its fluctuations not only measure the outcomes of the pricing strategy but also serve as a key indicator of its effectiveness. Accordingly, the state spaces of LRS and DPL are defined with explicit consideration of these two factors. The state space of SP v_i under DPL is defined as $S_{v_i} = \{C_{v_i}(t), p_{v_i}(t)\}$, where $C_{v_i}(t) \in [o_{v_i}, e_{v_i}]$ represents the capital value of SP v_i, and $p_{v_i}(t) \in [p_{v_i}^{min}, p_{v_i}^{max}]$ denotes the current service price. Here, o_{v_i} is the minimum capital threshold required for the SP to sustain operations within the system, while e_{v_i} represents the capital threshold corresponding to the SP's ability to reproduce new entities. The parameters $p_{v_i}^{min}$ and $p_{v_i}^{max}$ denote the lower and upper bounds of p_{v_i}, respectively. The state space of SP v_i under LRS is defined as $S_{v_i} = \{C_{v_i}(t)\}$. The current price is excluded from this state space, as the price state does not influence the selection of pricing strategies.

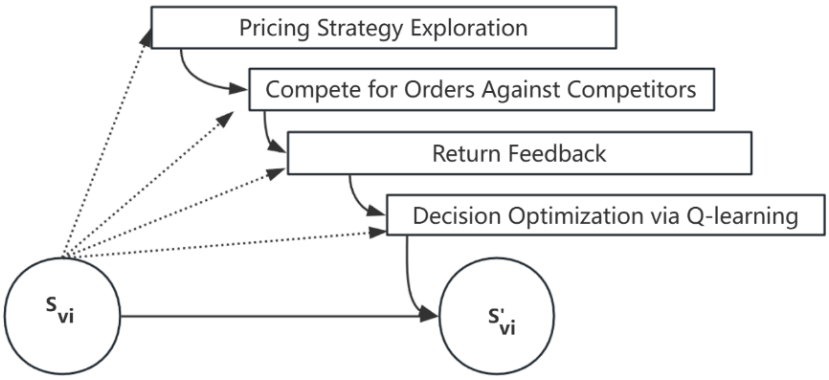


Figure 2. State transition process of a learning-enabled SP.

To reduce the complexity of state spaces, both $[o_{v_i}, e_{v_i}]$ and $[p_{v_i}^{min}, p_{v_i}^{max}]$ are discretized. Specifically, $[o_{v_i}, e_{v_i}]$ is divided into l equal-range intervals, while $[p_{v_i}^{min}, p_{v_i}^{max}]$ is divided into l' equal-range intervals. Each divided interval corresponds to a distinct state. We determine the corresponding state by identifying the interval to which the values of $C_{v_i}(t)$ and $p_{v_i}(t)$ belong.

4.2.2. Action space definition

The action space for SP v_i with LRS is defined as $A_{v_i} = \{a_{p_1}, a_{p_2}, a_{p_3}, a_{p_4}\}$. Here, a_{p_z} denotes the action of selecting the z-th pricing strategy from the set of four predefined rule-based strategies,

where $z = 1$ means CPP, $z = 2$ means SROP, $z = 3$ means SCSP, and $z = 4$ means MAPP. The action space for SP v_i with DPL is defined as $A_{v_i} = \{a_{up}, a_{down}\}$, where a_{up} represents the action of increasing the service price, and a_{down} means the action of decreasing it. The step sizes for price increases and decreases are set to be consistent with those of SROP, so as to better reflect the impact of learning capability on the final performance of pricing strategies.

4.2.3. State transition rule and reward function design

The $\varepsilon-$greedy strategy is a common approach for balancing exploration and exploitation. This section primarily employs the $\varepsilon-$greedy strategy to construct the action selection method. Specifically, at each decision point, a learning-enabled SP selects a random action with probability ε (exploration), creating an opportunity to discover potentially better strategies, and selects the current known optimal action with probability $1-\varepsilon$ (exploitation) to maximize immediate returns. As learning progresses, ε decays exponentially according to $\varepsilon_{t+1} = \eta * \varepsilon_t$, where η is the decay rate.

For a learning-enabled SP v_{i} at time t in state s_{v_i} ($s_{v_i} \in S_{v_i}$), the probability of selecting action a_{v_i} ($a_{v_i} \in A_{v_i}$) is calculated using Equation (5):

$$p(a_{\mathrm{v_i}}|s_{\mathrm{v_i}}) = \begin{cases} \dfrac{\varepsilon}{|A(s_{\mathrm{v_i}})|} + (1-\varepsilon), & \text{if } a_{\mathrm{v_i}} = \arg\max_{a'} Q(s_{\mathrm{v_i}}, a') \\ \dfrac{\varepsilon}{|A(s_{\mathrm{v_i}})|}, & \text{else} \end{cases} \tag{5}$$

After an action is selected, the Q-table is updated based on the reward R_t obtained by the SP from transactions with service demand in period *t*. The update follows Equation (6):

$$Q(s_{\mathrm{v_i}}, a_{v_i}) = (1-\alpha)Q(s_{\mathrm{v_i}}, a_{v_i}) + \alpha\left(R_t + \gamma \max_{a'_{v_i}} Q(s'_{v_i}, a'_{v_i})\right) \tag{6}$$

The reward function is calculated using Equation (7):

$$R_t = \sum_k f_{v_i,k}^{income} - f_{v_i}^{cost} \tag{7}$$

Here, $\sum_k f_{v_i,k}^{income}$ and $f_{v_i}^{cost}$ represent the total revenue earned and the total cost incurred by the SP during transaction period t, respectively.

5. Multi-agent simulation model for service recommendation

The effectiveness of different pricing strategies can be accurately evaluated only when they are deployed in dynamic and competitive service recommendation environments. Service recommendation is inherently complex, involving multiple entities with intricate interactions. Multi-agent simulation plays a crucial role in this evaluation by enabling the creation of an experimental platform that simulates service recommendation processes. This is accomplished by quantitatively modeling the fundamental attributes, decision-making behaviors, and interaction mechanisms of the entities involved.

5.1. SP agent design

Each SP agent represents an autonomous decision-making entity capable of dynamically adjusting its service pricing based on its internal state and the market environment. The specific design of agent attributes and decision-making behaviors (e.g., price update, agent demise, and reproduction) is described as follows:

(1) Agent attribute design

Except for the basic service information, the pricing strategy adopted, unit service cost, periodic sale volume, and capital value are the fundamental attributes of an SP agent. The service information of an agent for each transaction period t ($t \in T$) is defined as a tuple: $B_i(t) =< b_{i1}(t), b_{i2}(t), b_{i3}, b_{i4} >$. Here, $b_{i1}(t)$ represents the available service capacity, dynamically updated in each trading period based on transaction outcomes; $b_{i2}(t)$ denotes the price per unit service capacity, which is dynamically updated according to the agent's adopted pricing strategy; b_{i3} means the minimum delivery time of service, and b_{i4} signifies the service reliability—both of which are constants. Specifically, the update of $b_{i1}(t)$ follows the rule: $b_{i1}(t+1) = b_{i1}(t) - q_i(t)$, and $b_{i1}(t) = b_{i1}(1)$ every τ trading period [14]. Here, $q_i(t)$ is the sales volume of the SP agent in trading period t , calculated by Equation (8):

$$q_i(t) = \sum_{j \in V_i(t)} \varphi_j d_{j1}(t) \tag{8}$$

where $d_{j1}(t)$ denotes the required service capacity of demander u_j, $j \in V_i(t)$, $V_i(t)$ represents the set of SDs that select SP v_i for services, and φ_j means the minimum capacity delivery ratio allowed by SD u_j.

(2) Agent decision-making behavior modeling

The behaviors of an SP agent include capital settlement, price adjustment, and service capacity update. At the end of each period t, an SP agent calculates its revenue according to the profit function: $\pi_i(\mathrm{t}) = (b_{i2}(\mathrm{t}) - b_{ic}) * q_i(t)$, where b_{ic} represents the unit cost of service capacity. Following the calculation, the agent allocates a portion of its funds to cover normal operating expenses for the next transaction period [47], and updates its capital as: $C_{v_i}(t+1) = C_{v_i}(t) + \pi_i(t) - \bar{c}_{vi}$, where $\bar{c}_{vi}$ denotes the capital consumption of SP v_i for the subsequent period. After capital settlement, the agent adjusts its service price based on its selected pricing strategy.

The demise, survival, and reproduction of an agent is determined by its capital value: An agent will be eliminated if its capital falls below a predefined threshold (i.e., o_{v_i}). It will survive if its capital remains within the range $[o_{v_i}, e_{v_i}]$, and will reproduce a new offspring agent with an initial capital of $c_{initial}$ if its capital exceeds the upper limit threshold (i.e., e_{v_i}). The capital of the agent after reproduction is reduced to $c_{v_i}(t) - c_{\text{initial}}$ [48]. Moreover, refer to Xue et al. [19], where a fixed number of new SPs, denoted as r_1, enter the platform during each transaction period.

5.2. SD agent design

Each SD agent selects the most suitable SP based on service recommendation results and its own demand preferences. These agents interact with SP agents to complete transactions. The specific design of agent attributes and decision-making behaviors is described as follows:

(1) Agent attribute design

The attributes of an SD agent include the basic service demand information and the minimum capacity delivery ratio allowed (i.e., φ_j). Agent's service demand information is defined as a tuple: $D_j =< d_{j1}, d_{j2}, d_{j3}, d_{j4} >$, where d_{j1} represents the required service capacity, d_{j2} denotes the expected unit price the demander is willing to pay, d_{j3} refers to the expected service delivery time, and d_{j4} signifies the expected service reliability.

(2) Agent decision-making behavior modeling

The behaviors of an SD agent include demand creation, service evaluation, and service selection. The SD agent evaluates the utility of each candidate SP agent $v_i(v_i \in V_i(t))$ recommended by the platform. The utility function is defined as $f(B_i) = w_1\bar{b}_{i1}(t) + w_2(1 - \bar{b}_{i2}(t)) + w_3(1 - \bar{b}_{i3}) + w_4\bar{b}_{i4}$, where w_k, $k = 1, \ldots, 4$, represents the SD agent's preference weight for the k-th service attribute, and $w_1 + w_2 + w_3 + w_4 = 1$. The SD agent selects the SP agent with the highest utility for transaction.

Drawing on the SD arrival mechanism proposed by Xue et al. [19], a fixed number of SD agents, denoted as n_1, enter the platform during each transaction period. An SD agent's exit is determined by its demand-fulfillment status: if its demand is satisfied, the agent remains in the market and generates a new demand; if its demand remains unsatisfied for τ consecutive periods, the agent exits the market.

5.3. SRE agent design

The SRE agent functions as an intermediary connecting SP agents and SD agents. It processes and manages both service demand information and service attribute information, while also generating service recommendations. For each incoming service demand D_j, the SRE computes the similarity metrics $Sim(D_j, B_i)$ between D_j and the available manufacturing services B_i, and produces a corresponding recommendation list $V_j'(t)$. This recommendation information is then transmitted to the SD agent u_j for evaluation and selection. Furthermore, during the service contract-signing stage, the SRE aggregates the selection outcomes submitted by SD agents and forwards this feedback to the corresponding SP agents, thereby completing the transaction cycle and ensuring the closed-loop operation of recommendation, selection, and execution within the platform.

5.4. Dynamic interactions among agents

The dynamic interactions among agents are illustrated in Figure 3, which can be delineated into the following seven steps:

Step 1: *System initialization*. Initialize a service recommendation system with a specified number of SP and SD agents. Define the attributes and behavioral rules for each agent according to the previous design and modeling processes. SP agents and SD agents publish their corresponding service and demand information (i.e., information sets B and D) on the SRE agent. Set transaction period $t = 1$.

Step 2: *Service recommendation*. If the demand set D is empty, turn to **Step 5**; otherwise, the SRE agent randomly selects a demand from an SD agent, calculates the similarity $Sim(D_j, S_i), \forall v_i \in V$, recommends all candidate SPs (i.e., set V_j') that meet the recommendation threshold to the SD agent, removes the selected demand from set D, and turns to **Step 3**.

Step 3: *Service selection***.** Based on its service preferences, an SD agent computes the utility for each SP in the candidate set V_j'. It selects the SP that satisfies the minimum delivery quantity $\varphi_j d_{j1}(t)$ and offers the highest utility.

Step 4: *Transaction completion and capital settlement***.** The SRE agent sends the result of SP selection to the corresponding SP agent and acts as an intermediary to facilitate the signing of service contracts between SD and SP agents. Subsequently, the SP agent conducts capital settlement, and turns to **Step 2**.

Step 5: *Service price updating*. SP agents adjust their service prices based on their selected pricing strategies. Each SP agent can select one pricing strategy from the six options: CPP, SROP, SCSP, MAPP, LRS, and DPL.

Step 6: *Service information updating and new service demand creation*. If $t < T$, service information from SP agents is updated and new service demands are created. The SP agents decide whether to reproduce new agents or exit the platform based on their capital value. The SD agents determine whether to exit the platform or continue publishing new service demands based on their demand satisfaction status. New SP agents and SD agents arrive at the platform as well, and turn to **Step 2**. If $t = T$, turn to **Step 7**.

Step 7: *Stop*. Stop the recommendation system and observe its evolutionary outcome.

We observe the market competitiveness of the six pricing strategies by monitoring the final surviving quantity and the corresponding capital value of SP agents that adopt different pricing strategies.

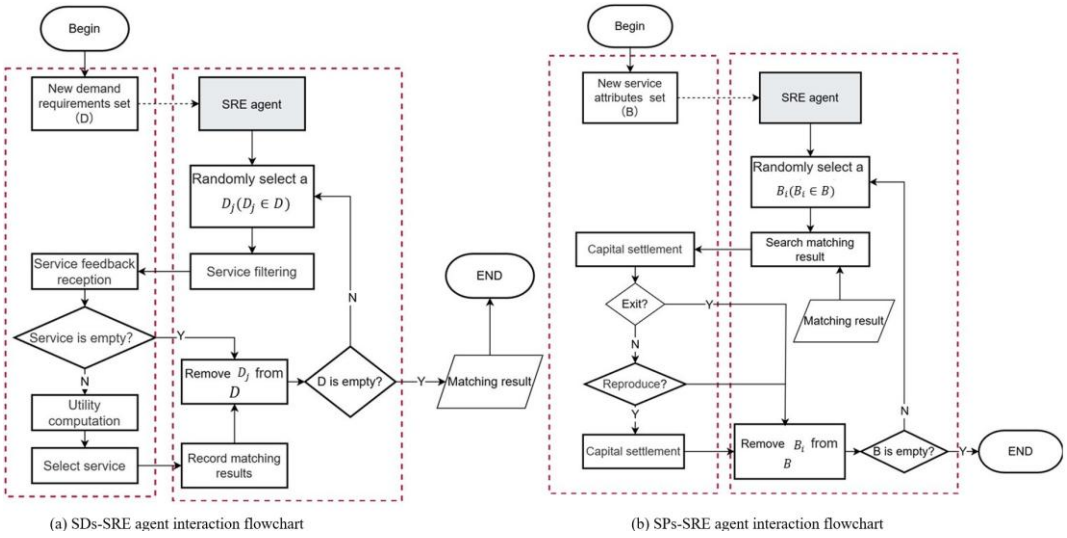


Figure 3. Agent interaction flowchart.

6. Experimental setup and simulation

6.1. Parameter settings

Yuncut is a manufacturing service platform headquartered in Jiaxing, China, that specializes in the optimal matching of services for steel plate cutting. The platform currently hosts over 1400 registered users and collaborates with more than 40 suppliers. To date, it has completed 3.75 million orders, and its annual revenue has grown substantially—from RMB 22 million in 2017 to RMB 149 million in 2024. To ensure efficient supply–demand matching in manufacturing services, Yuncut employs a service recommendation mechanism. Based on field research into Yuncut's service recommendation system, we collected and analyzed one week of real operational data. This statistical

analysis provided a basis for estimating the simulation parameter values used in this study. Among these parameters, b_{i1}, b_{i2}, b_{i3}, and b_{i4} are determined according to actual operational conditions, while parameters such as φ_j, n, and r are approximately estimated based on real-world scenarios. Detailed parameter values are shown in Table 1. We employ NetLogo 6.2 to develop the multi-agent system, and the visual interface is shown in Figure 4.

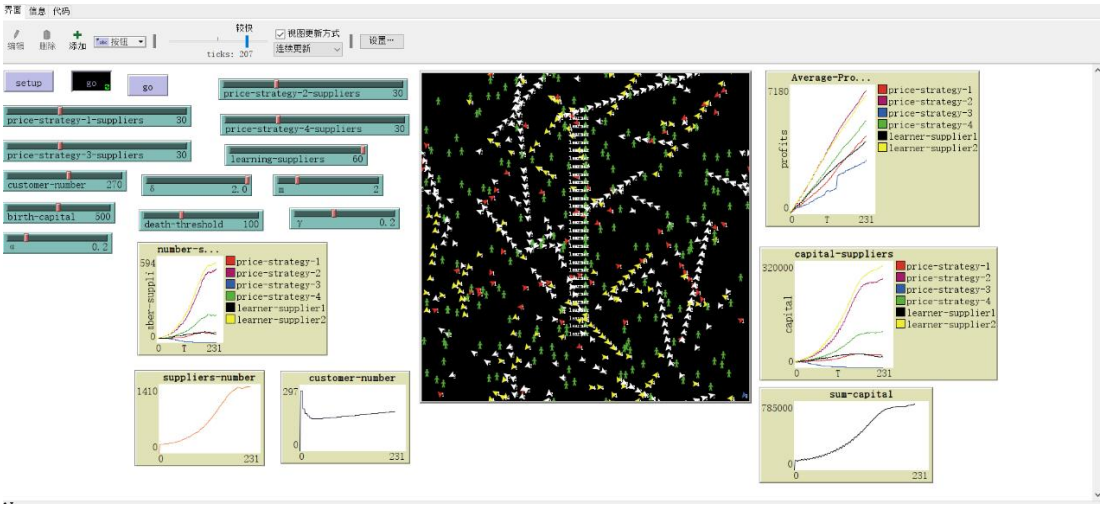


Figure 4. Multi-agent system simulation interface.

The simulation environment is configured as a 40×40 two-dimensional grid and includes six types of SP agents. Each type consists of 30 SP agents employing a distinct pricing strategy, resulting in a total of 180 SP agents. In addition, the environment comprises 90 SD agents (n = 90) and one SRE agent. All agents are initialized with randomly assigned positions and attribute values. Each experimental setup is repeated 10 times, and the average of the final results across these runs serves as the performance metric for evaluating different pricing strategies.

6.2. Benchmark experiments and model validity analysis

This study applies the event validity method to assess the effectiveness of the proposed model using a three-step procedure: (1) designing a baseline experiment, (2) generating output results through simulation runs, and (3) comparing the simulation outputs with real-world events. The baseline experiment was conducted using the parameter settings presented in Table 1, and the corresponding results are shown in Figures 5 and 6.

Table 1. Parameter settings.

Environment		SP agent		SD agent	
Parameter	Value	Parameter	Value	Parameter	Value
T	1000	c_{inital}	500	d_{j1}	[40, 100]
τ	5	$\bar{c}_{vi}$	[10, 20]	d_{j2}	[7.0, 10.0]
n	90	b_{i1}	[40, 100]	d_{j3}	[24, 96]
r	180	b_{i2}	[7.0, 10.0]	d_{j4}	[0.8, 1.0]
σ	2	b_{i3}	[24, 96]	φ_j	[0.8, 1.0]
n_1	1	b_{i4}	[0.8, 1.0]	w_i	[0.0, 1.0]
r_1	1	b_{ic}	[6.0, 7.0]		
l	90	e_{v_i}	1000	--	--
l'	3	o_{v_i}	100	--	--
α	0.2	β	0.2	--	--
γ	0.9	f_{in}	2	--	--
η	0.99995	$p_{v_i}^{min}$	7	--	--
ε	0.9	$p_{v_i}^{max}$	10	--	--

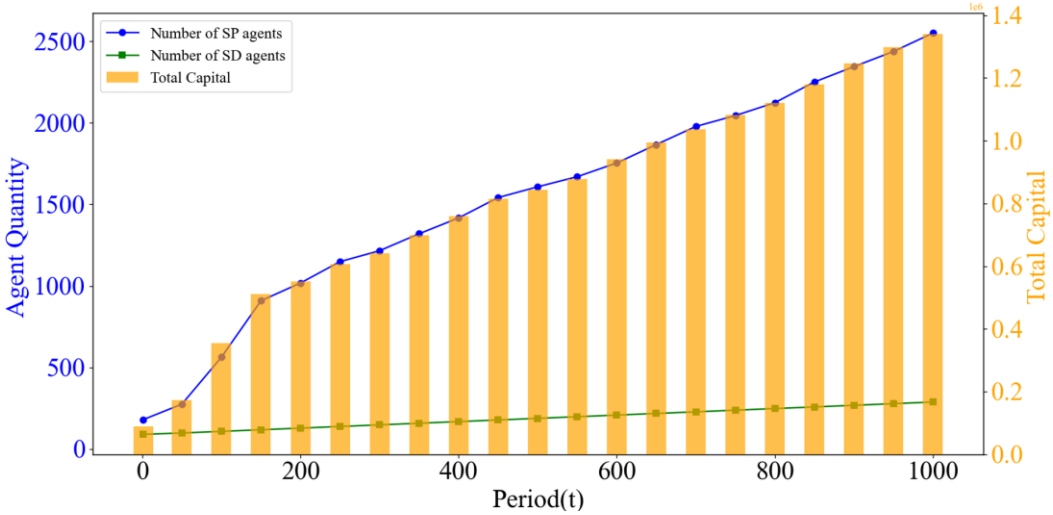
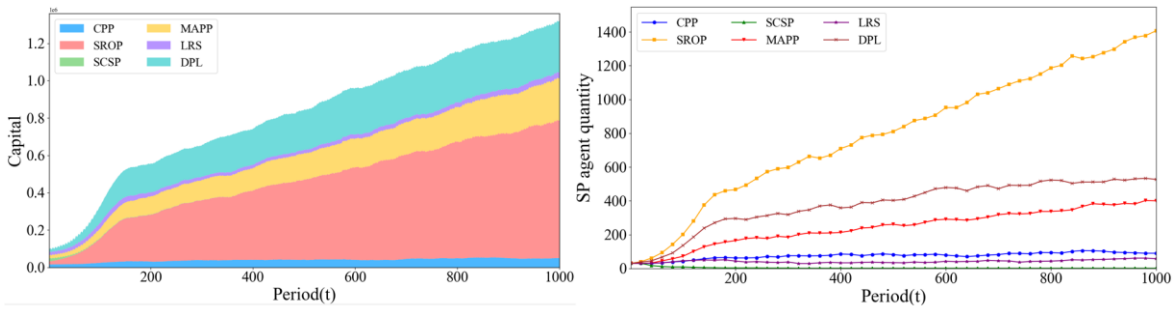


Figure 5. Evolutionary results of the recommendation system.



(a) Evolution of SP agents' capital (b) Evolution of SP agent quantity

Figure 6. Performance of different pricing strategies.

As shown in Figure 5, during the initial evolution phase, both the total number of SP agents and the overall market capital value increase rapidly. This pattern is consistent with the typical early-stage development dynamics of such platforms. As the simulation progresses, the growth in the number of SP agents and the total capital value gradually slows and eventually stabilizes, primarily due to the fixed incremental rate of SD agents. Service demand, as the core driver of the recommendation system, provides stable and continuous input that supports the ongoing expansion of SP participation and the accumulation of market capital value.

As illustrated in Figure 6, the evolutionary trajectories of SPs adopting different pricing strategies reveal substantial differences in survival rates and capital accumulation. Providers using the SROP strategy achieve both the highest survival levels and the strongest capital growth, underscoring the effectiveness of demand-oriented pricing within this ecosystem. In contrast, the MAPP and DPL strategies exhibit steady yet limited growth, suggesting constraints in exploiting service differentiation and inherent limitations in their price-based learning mechanisms. Although the LRS strategy faces similar challenges to DPL, its slower adaptation to market dynamics—particularly in the presence of SROP's first-mover advantage—further diminishes its competitiveness under increasing market pressure. Conversely, the CPP and SCSP strategies perform suboptimally. The former neglects market competition and demand elasticity, whereas the latter's narrow emphasis on supply-side factors weakens its capital conversion capability. Collectively, these findings highlight the critical importance of aligning pricing strategies with real-time demand signals in cloud manufacturing platforms.

6.3. Value setting of recommendation threshold δ

The recommendation threshold δ plays a critical role in shaping both the platform's selection of candidate SPs and the overall experience of SDs. To enhance the quality of recommendation lists and improve user experience by filtering out irrelevant service information, δ should be maintained within a moderate range. This section examines the impact of δ on the long-term development of the recommendation system. The experimental results are presented in Figure 7.

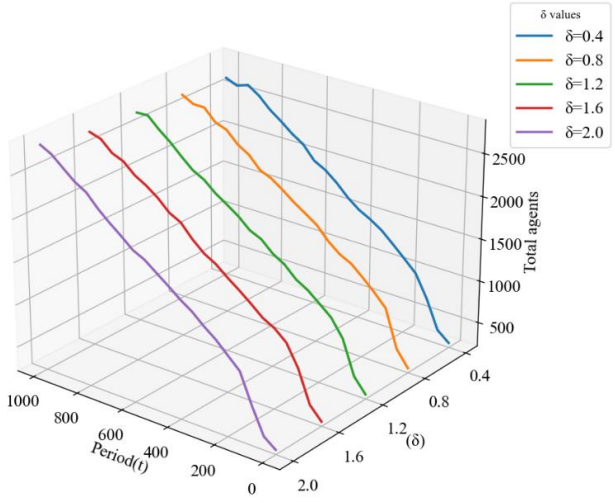


Figure 7. Evolution of total agents with variation in δ.

As shown in Figure 7, during the initial evolutionary phase ($t < 200$), the total number of agents increases rapidly in a jump-like manner, particularly in the case of $\delta = 0.4$. This effect arises because a lower δ value produces more concentrated service recommendations, allowing certain SPs to accumulate capital quickly and reach the predefined threshold. As the evolution progresses, the total number of agents shifts into a phase of steady growth. Notably, by the end of the evolutionary process, the scenario with $\delta = 2$ yields the highest total number of agents in the system. This result suggests that an excessively low δ may hinder the platform ecosystem's long-term development. Therefore, to more effectively observe the system's evolutionary dynamics, this study sets δ to 2.

6.4. Sensitivity analysis

6.4.1. Effects of capital upper limit threshold e_{v_i}

The capital upper-limit threshold e_{v_i} determines SP agents' decisions regarding the generation of new individuals. This section examines how variations in e_{v_i} influence the population dynamics of SP agents, with the experimental results shown in Figure 8.

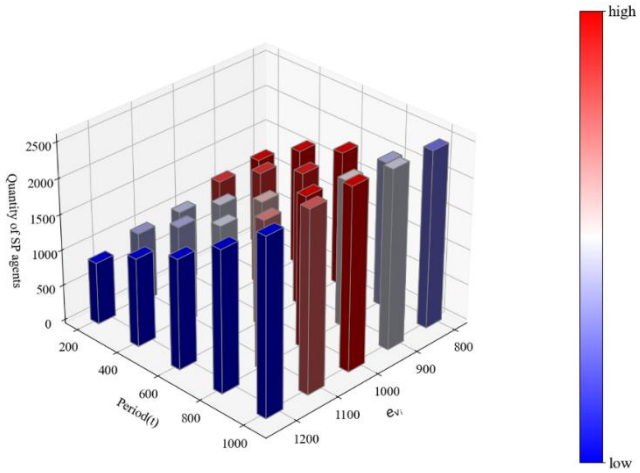
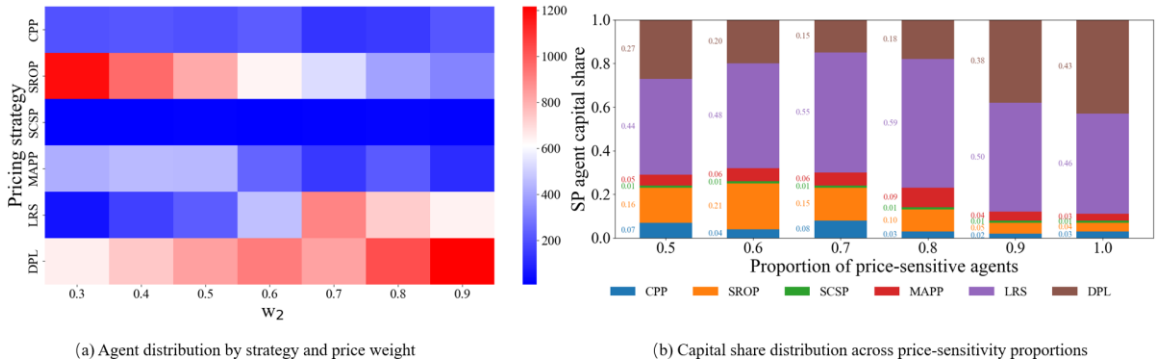


Figure 8. Evolution of SP agents' quantity with variation in e_{v_i}.

As illustrated in Figure 8, during the initial evolutionary phase ($t < 200$), the population of SP agents declines as the reproduction threshold (e_{v_i}) increases. However, this trend weakens as the system continues to evolve. By $t = 1000$, the relationship between e_{v_i} and the SP agent population no longer follows a simple linear pattern. The observed nonlinear dynamics arise from two underlying mechanisms. First, an excessively low e_{v_i} leads to rapid agent proliferation, which intensifies market competition and ultimately eliminates less competitive agents, thereby stabilizing population size. Conversely, an overly high e_{v_i} restricts successful reproduction for a significant subset of SP agents. These findings suggest that appropriately calibrating e_{v_i} is essential for sustaining an active and resilient SP agent population within the system.

6.4.2. Effects of price preference weight w_2

In practical application scenarios, demander preferences—acting as the primary driver of market demand—directly influence the effectiveness and market adaptability of different pricing strategies [49]. To enhance the practical relevance of this study, two price-sensitive SD scenarios are constructed. The first scenario investigates the impact of varying price weights (w_2) through seven comparative experiments, in which w_2 increases from 0.3 to 0.9 in increments of 0.1, while the remaining weights are allocated to the other three attributes. The second scenario assigns a fixed weight of 0.7 to the price attribute and 0.1 to each of the remaining attributes, and introduces six experimental groups based on the proportion of price-sensitive SDs on the platform (ranging from 50% to 100%). The evolutionary outcomes of these configurations are presented in Figure 9.



(a) Agent distribution by strategy and price weight

(b) Capital share distribution across price-sensitivity proportions

Figure 9. Effects of proportion of price-sensitive SDs.

Figure 9(a) shows that under low price-weight conditions ($w_2 < 0.5$), the dominant pricing strategies remain consistent with those observed in the benchmark experiment, indicating the broad applicability of the SROP strategy in such contexts. When w_2 increases to the range of 0.6–0.7, clear strategic differentiation emerges, demonstrating the substantial influence of price weight across all strategies. The performance of the SROP strategy declines markedly as w_2 rises, confirming its limited suitability in highly price-sensitive markets where price becomes the predominant decision factor, consequently diminishing the advantages of its composite decision-making mechanism. In contrast, the DPL strategy gains increasing advantage as w_2 grows, suggesting strong compatibility between its learning mechanism and price-sensitive market conditions. At $w_2 = 0.7$, the LRS strategy surpasses DPL to become the dominant approach. To further investigate the applicability boundaries of the LRS strategy, additional experiments were conducted, as presented in Figure 9(b).

Analysis of the dynamic evolution of SP agents' capital values reveals that providers employing the LRS strategy possess a pronounced competitive advantage (Figure 9(b)). This advantage becomes particularly evident when the proportion of price-sensitive demanders reaches 80%, at which point the strategy's learning mechanism aligns optimally with market conditions. Under these conditions, its learning effectiveness becomes markedly amplified, enabling LRS-based providers to outperform their competitors. SPs adopting the CPP, SCSP, and MAPP strategies exhibit comparatively lower and more stable capital value shares, reflecting consistent yet unremarkable performance in price-sensitive markets. A notable pattern emerges for the SROP strategy. When price-sensitive SDs constitute 60% of the population, SROP-based SPs achieve the second-highest capital share at 21%. However, as this

proportion increases to 100%, their share drops sharply to 4%, indicating that the competitiveness of SROP diminishes significantly as SD price sensitivity intensifies. In contrast, SPs utilizing the DPL strategy experience substantial gains: their capital share increases from 15% to 43% as the proportion of price-sensitive SDs rises from 70% to 100%. This trend demonstrates that in markets dominated by highly price-sensitive demanders, the learning mechanism embedded in the DPL strategy is highly effective, enabling these providers to adapt successfully to market shifts and secure a competitive edge.

6.4.3. Effects of risk coefficient β

The risk coefficient β, a key regulatory parameter in the SROP strategy, directly represents an SP's attitude toward pricing risk. This section investigates the impact of varying β values on the competitiveness of SROP. The coefficient was incrementally increased from 0.2 to 1.0 in steps of 0.2. The experimental results are presented in Figure 10.

The results in Figure 10 indicate that the population share of SPs adopting the SROP strategy initially increases and then decreases as the risk coefficient β rises, reaching a peak at $\beta = 0.4$. This suggests that a moderate risk level facilitates effective implementation of the strategy, whereas an excessively high risk ($\beta > 0.6$) diminishes its applicability. The underlying rationale is that a low risk coefficient limits the strategy's ability to balance risk and return effectively, while a high coefficient induces large price fluctuations and elevated risk, ultimately reducing the volume of recommendations.

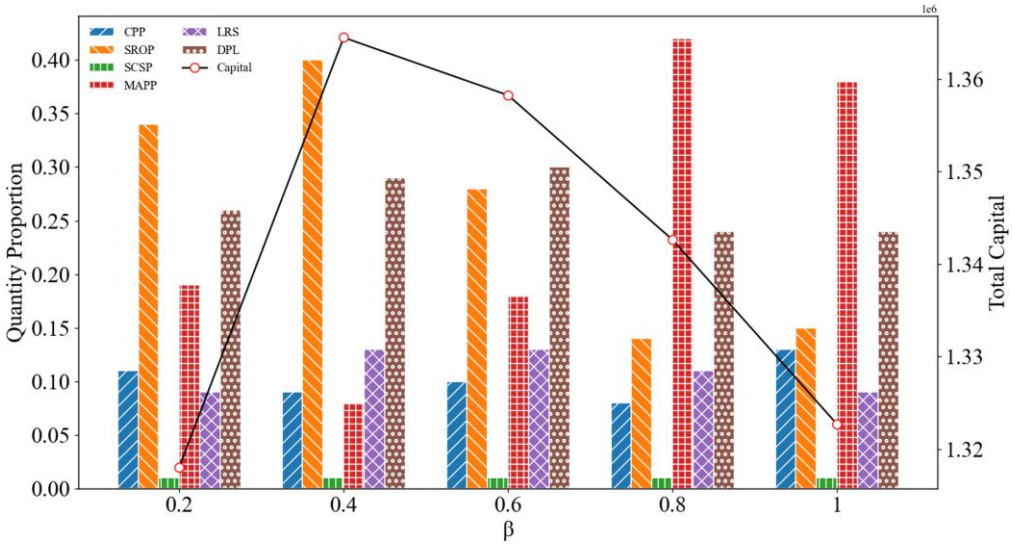


Figure 10. Effects of risk coefficient β.

In contrast, the population shares of SPs employing the CPP and SCSP strategies show no significant correlation with β, indicating that these strategies are less sensitive to risk factors. Notably, the population shares of both learning-enabled SPs are relatively high when β is 0.4 or 0.6, suggesting that a medium-risk environment is most favorable for the effectiveness of their learning mechanisms. Furthermore, when β exceeds 0.8, the MAPP strategy exhibits a marked advantage due to its pricing stability, implying that in environments characterized by aggressive price adjustments, a stable pricing strategy becomes more adaptive.

From a market-wide perspective, the total capital value initially increases and then declines as β rises, reaching a peak at $\beta = 0.4$. This indicates that the risk coefficient β exerts a significant influence on overall capital accumulation in the market. When β falls within an optimal range (e.g., 0.4–0.6), interactions among the various pricing strategies promote more rational resource allocation, allowing SPs to accumulate capital efficiently and thereby enhancing the total market capital value.

In summary, the SROP strategy exhibits a distinct advantage under low risk coefficients (β). As β increases to a medium range (0.4–0.6), the market experiences robust capital accumulation. During this phase, the market environment aligns well with the learning mechanisms of learning-enabled SPs, enabling them to capitalize on their strengths and promote capital growth. Conversely, at high β values, the pricing stability of the MAPP strategy proves particularly effective in managing complex market risks.

6.4.4. Effects of initial SD agent quantity

The initial number of SDs determines the order volume during the early stages of system evolution and ultimately affects service recommendation outcomes. To examine the impact of the initial SD count on the effectiveness of dynamic pricing strategies, three comparative experimental groups were designed, with initial SD numbers set at 90, 180, and 270. The experimental results are presented in Figure 11.

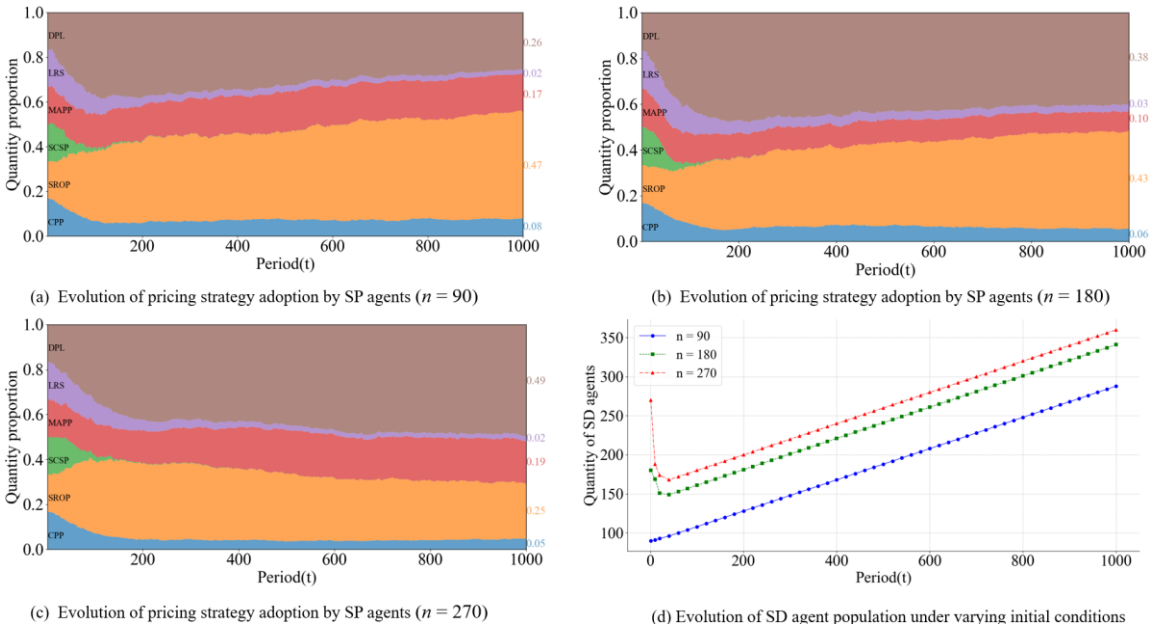


Figure 11. Effects of initial SD agent quantity.

As shown in Figure 11, the population share of SPs adopting the DPL strategy increases markedly with the initial number of SDs. Specifically, as the initial SD count rises from 90 to 270, the share of DPL-strategy SPs grows from 26% to 49%, establishing it as the dominant strategy. This phenomenon can be attributed to two underlying mechanisms. First, a sufficient order volume reduces market competition, providing a stable environment for SPs employing the DPL strategy. Second, the

diversification of SD demands exposes the limitations of traditional pricing strategies, further enhancing the relative advantage of DPL.

Figure 11(d) depicts the dynamic evolution of the SD population over time under different initial SD counts. In the scenario with a high initial SD count (n = 270), substantial SD churn occurs during the early stages of system evolution due to supply-demand imbalances. Nevertheless, driven by the platform's periodic SD entry mechanism, the total SD population eventually stabilizes and exhibits a steady growth trend. Notably, the magnitude of SD churn decreases as the initial SD count decreases. The final SD population size demonstrates a positive correlation with the initial SD count, indicating that the initial SD quantity significantly influences the system's evolutionary outcome.

In summary, the initial number of SDs exerts only a limited effect on the effectiveness of the CPP, SCSP, MAPP, and LRS strategies, whereas it has a pronounced impact on the performance of the SROP and DPL strategies. Moreover, as a critical prerequisite, the initial population of SD agents substantially determines the eventual scale of the platform's SD base.

7. Conclusions

The optimization of pricing strategies is crucial for SPs to secure more service recommendations. This paper designs four rule-based dynamic pricing strategies and further introduces reinforcement learning algorithms to construct two types of self-adaptive learning-driven dynamic pricing strategies. By integrating the relevant stakeholders involved in manufacturing service recommendation and the dynamic pricing decision-making process of SPs, a multi-agent simulation model is developed. The performance of various pricing strategies is analyzed across a range of simulated environments. The innovations and main research findings of this paper are summarized as follows:

(1) A multi-agent simulation model is developed to characterize the complex interactive behaviors among stakeholders. Using this model, the performance of the six dynamic pricing strategies can be evaluated by designing various artificial experiments based on the simulation results regarding the recommendation threshold δ and the capital upper limit threshold e_{v_i}. We find that while a higher δ yields marginal short-term benefits, it substantially enhances the platform's long-term sustainability. Conversely, an excessively low e_{v_i} inhibits both the expansion of the service provider population and market capital accumulation. Therefore, setting appropriate values for δ and e_{v_i} is crucial for the healthy development of the service recommendation system.

(2) Four rule-based pricing strategies are developed for SPs in platform-based manufacturing. Among them, the SROP strategy demonstrates a significant competitive advantage. In scenarios characterized by random user preferences and a limited initial scale of SDs, this strategy, through its adaptive price adjustment mechanism driven by real-time recommendation results and profit feedback, exhibits a more pronounced competitive edge compared to the other three rule-based dynamic pricing strategies.

(3) Two self-adaptive learning-driven pricing strategies are proposed for SPs considering the learning behaviors in competitive markets. The learning effectiveness of the two strategies is susceptible to various factors, such as the initial SD base, the distribution characteristics of SD price preferences, and the parameter settings of other rule-based pricing strategies. For instance, the competitive advantage of learning-driven pricing strategies becomes increasingly prominent as the

proportion of service price-sensitive SD rises. However, the effectiveness of learning-driven pricing strategies significantly decreases when the initial SD base is small and the initial market competition is intense. Consequently, if a service provider adopts such a strategy to enter the market, conducting a systematic assessment of the market environment is crucial.

Future research could focus on the development of more advanced, adaptive, learning-based dynamic pricing algorithms to further improve the market performance of such mechanisms. Additionally, the design of more sophisticated reward functions may enrich these algorithms by incorporating a broader range of decision-making factors beyond capital considerations.

Author contributions

Wenchong Chen: Conceptualization, Methodology, Writing – original draft, Funding acquisition. **Xiaoliao Tang:** Formal analysis, Data curation, Software, Visualization. **Jiehui Qi:** Investigation, Resources, Validation. **Hongwei Liu:** Supervision, Validation, Writing – review and editing. All authors reviewed and approved the final manuscript.

Use of Generative-AI tools declaration

AI tools used: ChatGPT (OpenAI, version [GPT-4])

Purpose of use: Assisted in data visualization design, code generation for figures, and language refinement.

Acknowledgments

The authors would like to thank Jinsong Zhong (Jiaxing Yun cut Online Technology Co., Ltd) for the kind help with providing real-world instances. This research was supported by the Philosophy and Social Science Planning Project of Zhejiang, China (Grant No. 23NDJC156YB).

Conflict of interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. T. Wang, C. Li, Y. Yuan, J. Liu, I. B. Adeleke, An evolutionary game approach for manufacturing service allocation management in cloud manufacturing, *Comput. Ind. Eng.,* **133** (2019), 231–240. https://doi.org/10.1016/j.cie.2019.05.005
2. H. Bo, Y. Zheng, Y. Li, S. He, Manufacturing resource outsourcing and matching: service mode selection and equilibrium evolution, *J. Oper. Res. Soc.,* **76** (2025), 772–789. https://doi.org/10.1080/01605682.2024.2392797

3. Q. Zhang, N. Li, J. Duan, J. Qin, Y. Zhou, Resource scheduling optimisation study considering both supply and demand sides of services under cloud manufacturing, *Systems,* **12** (2024), 133. https://doi.org/10.3390/systems12040133
4. H. Akbaripour, M. Houshmand, T. V. Woensel, N. Mutlu, Cloud manufacturing service selection optimization and scheduling with transportation considerations: mixed-integer programming models, *Int. J. Adv. Manuf. Technol.,* **95** (2018), 43–70. https://doi.org/10.1007/s00170-017-1167-3
5. H. Tong, J. Zhu, A customer-oriented method to support multi-task green scheduling with diverse time-of-use prices in Cloud Manufacturing, *Proc. Inst. Mech. Eng. B: J. Eng. Manuf.,* **237** (2023), 911–924. https://doi.org/10.1177/09544054221121848
6. W. Peng, W. Guo, L. Wang, R. Y. Liang, Dynamic pricing in cloud manufacturing systems under combined effects of consumer structure, negotiation, and demand, *Math. Probl. Eng.,* **2017** (2017), 2073585. https://doi.org/10.1155/2017/2073585
7. Y. S. Hao, Y. S. Fan, J. Zhang, Service recommendation based on description reconstruction in cloud manufacturing, *Int. J. Comput. Integr. Manuf.,* **32** (2019), 294–306. https://doi.org/10.1080/0951192X.2019.1571242
8. S. Zhang, W. Yang, S. Xu, W. Zhang, Control, A hybrid social network-based collaborative filtering method for personalized manufacturing service recommendation, *Int. J. Comput Commun Control.,* **12** (2017), 728–740. https://doi.org/0000-0002-6405-584X
9. L. Wang, T. Gao, B. Zhou, H. Tang, F. Xiang, Manufacturing service recommendation method toward industrial internet platform considering the cooperative relationship among enterprises, *Expert Syst. Appl.,* **192** (2022), 116391. https://doi.org/10.1016/j.eswa.2021.116391
10. Z. C. Liu, L. Wang, X. X. Li, S. Pang, A multi-attribute personalized recommendation method for manufacturing service composition with combining collaborative filtering and genetic algorithm, *J. Manuf. Syst.,* **58** (2021), 348–364. https://doi.org/10.1016/j.jmsy.2020.12.019
11. J. Liu, Y. Chen, Q. Liu, B. Tekinerdogan, A similarity-enhanced hybrid group recommendation approach in cloud manufacturing systems, *Comput. Ind. Eng.,* **178** (2023), 109128. https://doi.org/10.1016/j.cie.2023.109128
12. H. Bouzary, F. F. Chen, A classification-based approach for integrated service matching and composition in cloud manufacturing, *Rob. Comput. Integr. Manuf.,* **66** (2020), 101989. https://doi.org/10.1016/j.rcim.2020.101989
13. X. B. Zhu, J. Shi, F. J. Xie, R. Q. Song, Pricing strategy and system performance in a cloud-based manufacturing system built on blockchain technology, *J. Intell. Manuf.,* **31** (2020), 1985–2002. https://doi.org/10.1007/s10845-020-01548-3
14. X. Xue, S. F. Wang, L. J. Zhang, Z. Y. Feng, Evaluating of dynamic service matching strategy for social manufacturing in cloud environment, *Future Gener. Comput. Syst.,* **91** (2019), 311–326. https://doi.org/10.1016/j.future.2018.08.028
15. X. Xue, Y. M. Kou, S. F. Wang, Z. Z. Liu, Computational experiment research on the equalization-oriented service strategy in collaborative manufacturing, *IEEE Trans. Serv. Comput.,* **11** (2016), 369–383. https://doi.org/10.1109/TSC.2016.2569082
16. P. Cong, J. Zhou, M. Chen, T. Wei, Personality-guided cloud pricing via reinforcement learning, *IEEE Trans. Cloud Comput.,* **10** (2020), 925–943. https://doi.org/10.1109/TCC.2020.2992461

17. Y. P. Zhang, Y. Cheng, H. T. Zheng, F. Tao, Long-short-term preference based dynamic pricing and manufacturing service collaboration optimization, *IEEE Trans. Ind. Inf.,* **18** (2022), 8948–8956. https://doi.org/10.1109/tii.2022.3153663
18. A. Kastius, R. Schlosser, Dynamic pricing under competition using reinforcement learning, *J. Revenue Pricing Manag.,* **21** (2021), 50–63. https://doi.org/10.1057/s41272-021-00285-3
19. X. Xue, D. Y. Zhou, F. Y. Chen, X. N. Yu, Z. Y. Feng, Y. C. Duan, et al., From soa to voa: a shift in understanding the operation and evolution of service ecosystem, *IEEE Trans. Serv. Comput.* **16** (2021), 315–329. https://doi.org/10.1109/TSC.2021.3134718
20. M. Bisceglia, R. Cellini, L. Siciliani, O. R. Straume, Optimal dynamic volume-based price regulation, *Int. J. Ind Organiz.,* **73** (2020), 102675. https://doi.org/10.1016/j.ijindorg.2020.102675
21. G. Gallego, G. V. Ryzin, Optimal dynamic pricing of inventories with stochastic demand over finite horizons, *Manage. Sci.,* **40** (1994), 999–1020. https://doi.org/10.1287/mnsc.40.8.999
22. Y. Wu, P. Liu, Pricing strategies for shared manufacturing platform considering cooperative advertising based on differential game, *PLoS One,* **19** (2024), e0303928. https://doi.org/10.1371/journal.pone.0303928
23. R. Bi, F. Wu, S. Yuan, Leadership-driven pricing and customization in collaborative manufacturing: a platform dynamics perspective, *J. Theor. Appl. Electron. Commer. Res.,* **20** (2025), 222. https://doi.org/10.3390/jtaer20030222
24. L. Sun, G. Hua, T. Cheng, Y. Wang, How to price 3D-printed products? Pricing strategy for 3D printing platforms, *Int. J. Prod. Econ.,* **226** (2020), 107600. https://doi.org/10.1016/j.ijpe.2019.107600
25. F. Tao, Y. Zuo, L. D. Xu, L. Zhang, IoT-based intelligent perception and access of manufacturing resource toward cloud manufacturing, *IEEE Trans. Ind. Inf.,* **10** (2014), 1547–1557. https://doi.org/10.1109/TII.2014.2306397
26. X. V. Wang, X. W. Xu, An interoperable solution for cloud manufacturing, *Rob. Comput. Integr. Manuf.,* **29** (2013), 232–247. https://doi.org/10.1016/jscim.2013.01.005
27. X. Cao, H. Bo, Y. Liu, X. Liu, Effects of different resource-sharing strategies in cloud manufacturing: A stackelberg game-based approach, *Int. J. Prod. Res.,* **61** (2023), 520–540. https://doi.org/10.1080/00207543.2021.2010824
28. P. K. Kopalle, K. Pauwels, L.Y. Akella, M. Gangwar, Dynamic pricing: Definition, implications for managers, and future research directions, *J. Retailing.,* **99** (2023), 580–593. https://doi.org/10.1016/j.jretai.2023.11.003
29. O. Besbes, A. Zeevi, On the (surprising) sufficiency of linear models for dynamic pricing with demand learning, *Manage. Sci.,* **61** (2015), 723–739. https://doi.org/10.1287/mnsc.2014.2031
30. Y. Wu, X. Zhou, Q. Xia, L. Peng, Resource scheduling method for equipment maintenance based on dynamic pricing model in cloud manufacturing, *Appl. Sci.,* **13** (2023), 12483. https://doi.org/10.3390/app132212483
31. J. Zheng, Y. Gan, Y. Liang, Q. Jiang, J. Chang, Joint strategy of dynamic ordering and pricing for competing perishables with Q-Learning algorithm, *Wirel. Commun. Mob. Comput.,* **2021** (2021), 6643195. https://doi.org/10.1155/2021/6643195

32. M. Salb, L. Jovanovic, A. Elsadai, N. Bacanin, V. Simic, D. Pamucar, et al., Cloud spot instance price forecasting multi-headed models tuned using modified PSO, *J. King Saud Univ. Sci.,* **36** (2024), 103473. https://doi.org/10.1016/j.jksus.2024.103473
33. A. Guizzardi, F. M. E. Pons, G. Angelini, E. Ranieri, Big data from dynamic pricing: A smart approach to tourism demand forecasting, *Int. J. Forecast.,* **37** (2021), 1049–1060. https://doi.org/10.1016/j.ijforecast.2020.11.006
34. Q. Ma, S. Feng, J. Liu, Dynamic pricing and demand forecasting: Integrating time-series analysis, regression models, machine learning, and competitive analysis, *Appl. Comput. Eng.,* **93** (2024), 149–154. https://doi.org/10.54254/2755-2721/93/20240935
35. C. Zhao, L. Zhang, Y. Liu, Z. Zhang, G. Yang, B. H. Li, Agent-based simulation platform for cloud manufacturing, *Int. J. Model. Simul. Sci. Comput.,* **8** (2017), 1742001. https://doi.org/10.1007/s00170-015-7221-0
36. C. Zhao, X. Luo, L. Zhang, Modeling of service agents for simulation in cloud manufacturing, *Rob. Comput. Integr. Manuf.,* **64** (2020), 101910. https://doi.org/10.1016/j.rcim.2019.101910
37. Y. Hu, L. Pan, D. Gu, Z. Wang, H. Liu, Y. Wang, Matching of manufacturing resources in cloud manufacturing environment, *Symmetry,* **13** (2021), 1970. https://doi.org/10.3390/sym13101970
38. J. Zhang, C. Wang, Supply–demand dynamic matching in cloud manufacturing based on hypernetwork model, *Appl. Sci.,* **15**, (2025), 1747. https://doi.org/10.3390/app15041747
39. L. Guo, S. Wang, L. Kang, Y. Cao, Agent-based manufacturing service discovery method for cloud manufacturing, *Int. J. Adv. Manuf. Technol.,* **81** (2015), 2167–2181. https://doi.org/10.1007/s00170-015-7221-0
40. K. Li, T. Zhou, B. H. Liu, H. Li, A multi-agent system for sharing distributed manufacturing resources, *Expert Syst. Appl.,* **99** (2018), 32–43. https://doi.org/10.1016/j.eswa.2018.01.027
41. H. Bouzary, F. F. Chen, K. Krishnaiyer, Service matching and selection in cloud manufacturing: a state-of-the-art review, *Proc. Manuf.,* **26** (2018), 1128–1136. https://doi.org/10.1016/j.promfg.2018.07.149
42. A. Simeone, Y. Zeng, A. Caggiano, Intelligent decision-making support system for manufacturing solution recommendation in a cloud framework, *Int. J. Adv. Manuf. Technol.,* **112** (2021), 1035–1050. https://doi.org/10.1007/s00170-020-06389-1
43. F. Fkih, Similarity measures for collaborative filtering-based recommender systems: review and experimental comparison, *J. King Saud. Univ. Comput. Inf. Sci.,* **34** (2022), 7645–7669. https://doi.org/10.1016/j.jksuci.2021.09.014
44. A. Dolgui, J. M. Proth, Pricing strategies and models, *Annu. Rev. Control.,* **34** (2010), 101–110. https://doi.org/10.1016/j.arcontrol.2010.02.005
45. X. Chang, J. Li, D. Rodriguez, Q. Su, Agent-based simulation of pricing strategy for agri-products considering customer preference, *Int. J. Prod. Res.,* **54** (2016), 3777–3795. https://doi.org/10.1080/00207543.2015.1120901
46. I. Dogan, A. R. Güner, A reinforcement learning approach to competitive ordering and pricing problem, *Expert Syst. Appl.,* **32** (2015), 39–48. https://doi.org/10.1111/exsy.12054
47. A. Bhanu, G. C. Nath, T. Patnaik, Unlocking liquidity through shortened settlement cycle: Empirical evidence from India, *Econ. Lett.,* **239** (2024), 111736. https://doi.org/10.1016/j.econlet.2024.111736

48. X. Xue, Z. J. Chen, S. F. Wang, Z. Y. Feng, Y. C. Duan, Z. B. Zhou, Value entropy: A systematic evaluation model of service ecosystem evolution, *IEEE Trans. Serv. Comput.,* **15** (2020), 1760–1773. https://doi.org/10.1109/TSC.2020.3016660
49. C. Panico, C. Cennamo, User preferences and strategic interactions in platform ecosystems, *Strategic Manage. J.,* **43** (2022), 507–529. https://doi.org/10.1002/smj.3149

AIMS Press