

LLM-empowered Agents Simulation Framework for Scenario Generation in Service Ecosystem Governance

Deyu Zhou^{1,2}, Yuqi Hou¹, Xiao Xue^{3,4,5*}, Xudong Lu^{1,2}, Qingzhong Li^{1,6}, Lizhen Cui^{1,2*}

¹ School of Software, Shandong University, Jinan, China

² Joint SDU-NTU Centre for Artificial Intelligence Research (C-FAIR), Shandong University, Jinan, China

³ College of Intelligence and Computing, Tianjin University, Tianjin, China

⁴ Tianjin Key Laboratory of Healthy Habitat and Smart Technology, Tianjin, China

⁵ Laboratory of Computation and Analytics of Complex Management Systems, Tianjin University, Tianjin, China

⁶ ShanDa Dareway Software Co. Ltd, Jinan, China

Email: zhoudedyu@mail.sdu.edu.cn, 202100300341@mail.sdu.edu.cn, jzxuexiao@tju.edu.cn, dongxul@sdu.edu.cn, lqz@sdu.edu.cn, clz@sdu.edu.cn

Abstract—As the social environment is growing more complex and collaboration is deepening, factors affecting the healthy development of service ecosystem are constantly changing and diverse, making its governance a crucial research issue. Applying the scenario analysis method and conducting scenario rehearsals by constructing an experimental system before managers make decisions, losses caused by wrong decisions can be largely avoided. However, it relies on predefined rules to construct scenarios and faces challenges such as limited information, a large number of influencing factors, and the difficulty of measuring social elements. These challenges limit the quality and efficiency of generating social and uncertain scenarios for the service ecosystem. Therefore, we propose a scenario generator design method, which adaptively coordinates three Large Language Model (LLM) empowered agents that autonomously optimize experimental schemes to construct an experimental system and generate high quality scenarios. Specifically, the Environment Agent (EA) generates social environment including extremes, the Social Agent (SA) generates social collaboration structure, and the Planner Agent (PA) couples task-role relationships and plans task solutions. These agents work in coordination, with the PA adjusting the experimental scheme in real time by perceiving the states of each agent and these generating scenarios. Experiments on the ProgrammableWeb dataset illustrate our method generates more accurate scenarios more efficiently, and innovatively provides an effective way for service ecosystem governance related experimental system construction.

Index Terms—Scenario Generator, LLM-empowered Agents, Service Ecosystem, Social Simulation, Experimental Design

I. INTRODUCTION

As the social landscape is redefined under the service logic, the service system has been gradually transcending the boundaries of traditional industries and evolving into a service ecosystem that is collaboratively operated through the division

* Corresponding author: Lizhen Cui and Xiao Xue; e-mail: clz@sdu.edu.cn; jzxuexiao@tju.edu.cn

of labor among diverse intelligent service entities, including individuals, communities, intelligent robots, and AI platforms.

In this novel service system, various entities, namely service providers, service consumers, form a dynamic collaboration network via resource virtualization and servitization, thereby enabling cross-industry value co-creation (the right part of Fig.1). Its operation mechanism is characterized by three prominent features. Firstly, social demands exhibit a dual nature of dynamism and complexity. The evolution of consumer preferences and market fluctuations interactively propel the rapid reconstruction of service scenarios. Secondly, there exists a significant heterogeneity in the intelligence of entities. The emotional cognition of humans and the rational decision-making of AI give rise to a multi-dimensional interaction pattern. Thirdly, the system connections yield non-linear effects, where a strategic adjustment of a single node has the potential to trigger a global cascading reaction [1]. These features render the service ecosystem a typical complex adaptive system, posing severe challenges to traditional governance methods when confronted with scenarios such as abrupt changes in demand and the failure of entity collaboration [2].

Currently, research on the governance of service ecosystems predominantly employs three methodologies: data-driven modeling and anomaly monitoring, macro-evolutionary analysis of complex networks, and experimental simulation based on Agent-based Modeling (ABM) [3]. Despite the progress achieved by these methods in specific scenarios, they share three common limitations.

- The construction of the experimental system relies on predefined rules, making it arduous to model the heterogeneity of entity intelligence and the dynamics of social relationships.
- There is an insufficiency in the generation of extreme scenarios. Existing data-driven methods are constrained

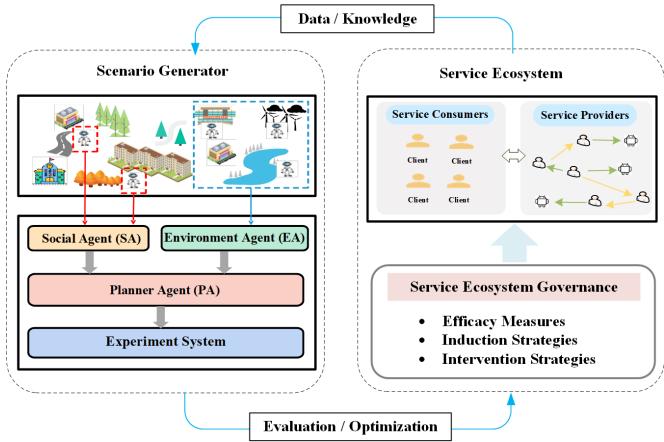


Fig. 1. Relationship between scenario generator and service ecosystem. Based on the data and knowledge from service ecosystem, scenario generator collaboratively generates experimental schemes by PA, SA and EA. Then, the experiment system produces a series of scenarios to assist in system governance and optimization.

by the distribution of historical data, resulting in limited coverage of “black swan” events.

- The complexity of the system means that parameter calibration requires a large amount of human resources and time, making it difficult to adapt to the real-time evolution of the system promptly.

These issues have led to deficiencies such as scenario distortion and decision-making delays in existing governance strategies, necessitating the exploration of a new methodological framework. Therefore, this paper proposes a scenario generator design method driven by LLM-empowered agents, a core component of the experimental system. The framework integrates three collaborative agents (i.e., the left part of Fig.1): **Planner Agent (PA)** dynamically compiles domain rules via RAG and program-aided prompting, translating natural language constraints into executable logic. **Environment Agent (EA)** employs LLM-based semantic deconstruction and adversarial prompt engineering to generate extreme scenarios (e.g., policy shifts, demand surges), overcoming traditional statistical constraints. **Social Agent (SA)** models heterogeneous entities’ behavioral strategies by distilling social relationship networks through LLM-driven cognitive simulations.

The scenario generator synthesizes scenarios by interacting with the data knowledge base from service ecosystem [4]. These scenarios undergo experimental deduction, yielding decision insights to optimize the service ecosystem’s matching logic and operator strategies. The closed-loop “generate-validate-optimize” mechanism enables adaptive governance of complex service ecosystems under uncertainty. Experiments demonstrate significant improvements in scenario generation efficiency and feature coverage, offering a novel tool for digital-era service ecosystem governance [5].

The academic contributions of this paper are manifested in the following aspects:

- Establishing a scenario generation framework based on

the collaboration of multiple agents of the LLM, breaking through the paradigm of predefined rules through semantic-driven approaches.

- Proposing a backbone extraction method for the collaborative relation of heterogeneous agents, enabling the quantitative characterization of the level of human intelligence and correlation relationships by utilizing the social cognitive simulation capabilities of the LLM.
- Constructing an adversarial generation-verification closed loop for extreme scenarios, enhancing the coverage of long-tail events through the causal reasoning capabilities of the LLM.

The rest of this paper is organized as follows: Section II presents the research background and motivation; Section III presents the modeling framework of the scenario generator; Section IV gives the design and analysis of computational experiments; Section V discusses and concludes the paper.

II. BACKGROUND AND MOTIVATION

Influencing factors in the evolution process of service ecosystem are complex and difficult to confirm. They are not only affected by human free will and subjective initiative, but sometimes accidental events may also change the evolution path, posing huge challenges to the governance of service ecosystems [6]. The commonly used analysis methods currently used by researchers can be summarized into the following three types:

1) Data-driven analysis identifies potential system failures by monitoring operational data, detecting anomalies like outliers, fluctuation points, and abnormal time series [7]. These methods include statistical models (e.g., hypothesis testing), proximity-based techniques (e.g., KNN), and density-based approaches (e.g., LOF). Recent advances emphasize multimodal data integration: Hongyi Liu et al. proposed an adversarial contrastive learning framework to fuse metrics and logs, improving anomaly detection robustness in microservices [8]; Mingyi Liu et al. designed a multilayer network model capturing cross-layer interactions (functionality, collaboration, competition) to predict system evolution and enable multi-granularity governance [9].

2) Complex network analysis effectively characterizes service ecosystem structures and evolution. Shin et al. used centrality metrics (degree, betweenness) and clustering to identify research trends from 340 publications [10]. Wang Z. and Tu Z. proposed a service community framework to detect historical patterns (e.g., cluster differentiation, strategy convergence) for trend prediction [11]. Huang Keman et al. applied this approach to web service ecosystems, quantifying static topologies (node connectivity) and dynamic behaviors (service lifecycles) [12]. Ali Yassin et al. developed netbone, a Python package evaluating backbone extraction techniques in weighted networks, covering statistical, structural, and hybrid methods [13]. While powerful for macro-level analysis, these methods depend on predefined topologies and overlook micro-level interactions.

3) Agent-based modeling addresses this gap by simulating autonomous agents to uncover system-level evolution mechanisms. Villalba et al. emphasized self-organization, self-adaptability, and long-lasting evolvability in service systems, validating designs through multi-agent simulations [14]. Ye et al. modeled services as autonomous agents, developing an agent-based self-evolving composition method covering five stages: discovery, selection, negotiation, execution, and adaptation [15]. Lim et al. designed the AppEco model to simulate mobile app ecosystems, evaluating development strategies' impacts on downloads, diversity, and adoption rates [16]. Xue et al. further introduced a social learning framework to model competitive-cooperative evolution across individual, organizational, and societal layers [17], [18].

In summary, current research on service ecosystems predominantly focuses on the statistical analysis and replication of macro-level system characteristics, which heavily relies on researchers' a priori knowledge and assumptions about the system [19]. Three critical challenges persist:

- Generating extreme scenarios and long-tail distribution events to address "black swan" risks that traditional data-driven approaches fail to capture;
- Quantifying qualitative features such as the diverse intelligence levels of heterogeneous agents and complex social relationship networks, thereby transcending the limitations of predefined rules;
- Achieving low-cost parameter calibration and real-time evolutionary adaptation to reduce experimental costs while ensuring the dynamic optimization of complex service ecosystems.

Recent advances in LLM technology and scenario analysis methods have expanded service ecosystem governance opportunities [20]. Scenario analysis, formalized by Pierre Wack in 1970s, systematically evaluates system dynamics to design plausible futures through narrative-driven frameworks. Qualitative approaches rely on expert knowledge and visual tools (e.g., diagrams) to map factor relationships and integrate insights into future projections [21], [22]. While modern tools enhance data visualization and simulation (e.g., Sora's text-to-scene generation), their output quality depends heavily on scenario descriptions. Quantitative methods, like scenario generators, automate parameter-based systems (e.g., autonomous driving [23], [24], renewable energy systems [25]) by abstracting problems into models with predefined distributions. However, they struggle with complex social systems due to oversimplification and inability to capture multi-element interactions.

Relying on LLM technology, this paper applies the scenario analysis method to the research on the governance of service ecosystems and proposes the scenario generator design method for service ecosystem governance driven by LLM empowered agents simulation to address the above challenges.

III. DESIGN METHODOLOGY OF THE SCENARIO GENERATOR FRAMEWORK

This section initially expounds upon the overarching framework of the scenario generator, as shown in Fig.2. Subsequently, a detailed elaboration is conducted for each module within the framework. Specifically, it involves the generation of extreme scenarios by the Environment Agent (EA), and during the generation process of the Social Agent (SA), the extraction of the backbone network pertaining to the subject-collaborative process. Ultimately, the rule - generation mechanism within the Planner Agent (PA) and the process of scheme calibration are expounded.

A. Problem Formulation

The scenario of a service ecosystem can be conceptualized as a holistic outcome derived from the interactions between the demand environment and service entities over a temporal interval T . This framework systematically delineates the constituent elements encompassing demand environment characteristics, operational status of service supply-side entities, interaction networks among stakeholders, along with response variables that reflect system efficacy states.

Within the time period T (T is the dimension of the scenario), and on a known probability space, the parameter vector $\varepsilon_t (t = 1, 2, \dots, T)$ is composed of random variables that follow a finite discrete distribution. Then $\tau = (\varepsilon_1, \varepsilon_2, \dots, \varepsilon_T)$ forms a matrix composed of a T -dimensional random vector. A realization value $\tau^k = (\varepsilon_1, \varepsilon_2, \dots, \varepsilon_{T_k})$ of τ is called a scenario. The scenario generation method is used to generate a set $S (S = \{\tau^1, \tau^2, \dots, \tau^n\})$ containing n scenarios. Each scenario represents a situation that may occur in reality and corresponds to a scenario occurrence probability p_s . The larger the scenario probability, the greater the possibility of occurrence. The sum of all scenario probabilities is 1. Considering comprehensive influencing factors (including the environment and the subject) and response variables, the parameter vector ε_t at time t is defined as follows:

$$\varepsilon_t(e_t, a_t, u_t) = (E(e_t), I(a_t), U(u_t))^T, \quad (1)$$

$$\tau^c = \min\{G_{Scenes}(S)|e, a, u\}, \quad (2)$$

$$X \rightarrow S, \forall \tau^k \in S, \tau^c \subset S \quad (3)$$

Here, $E()$ represents the method for normalizing the vector composed of the environment elements e ; $I()$ represents the method for normalizing the vector composed of the Agent collaborative network elements a ; $U()$ is the representation method of the system effectiveness u ; τ^c represents the scenario set finally generated by the algorithm, which contains c scenarios, and the sum of the probabilities of all scenarios occurring is less than or equal to 1; $G_{Scenes}(S)$ represents generating the minimum number of scenarios from the solution set S ; X represents the sample space used to generate the corresponding elements in the parameter vector ε_t ; \rightarrow indicates that the n -dimensional vector ε_t representing the state of each time - instant of each scenario τ^k in the scenario set S is

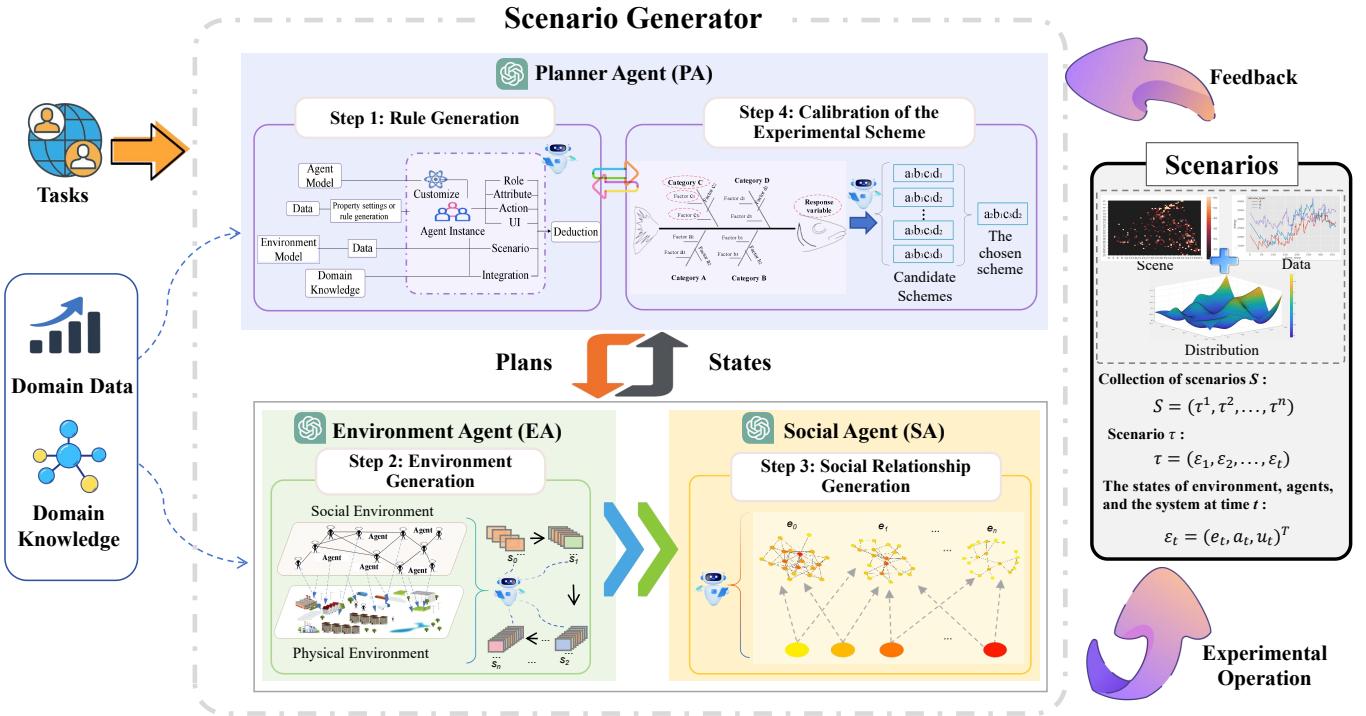


Fig. 2. The framework of scenario generator design method. Rules generated by PA, the environment model created by EA, and social relationships formed by SA are developed to experimental scheme. Finally, it runs to output a scenario set for experimental analysis, and the scheme is calibrated through feedback.

obtained by sampling from n value distributions in the set X respectively.

The system effectiveness u_t at time period t serves as the response variable of the system, which is used to reflect the indicator that researchers focus on, representing the overall state of the system at time period t . According to [26], Value Entropy is used to measure the overall health degree of the system. Assuming that the total number of individuals in the system is N_{total} , based on the calculation method of value entropy, the system effectiveness u_t at time period t is defined as shown in the following formula:

$$u_t = e^{1 - \left| \frac{-\sum_{j=1}^m p_j \log_2 p_j - \log_2 \sqrt{N_{total}}}{\log_2 \sqrt{N_{total}}} \right|} \quad (4)$$

$$p_j = \frac{N_j}{N_{total}} \quad (5)$$

Here, m represents the number of niches divided in the service ecosystem; N_j is the number of individuals in niche j . $\log_2 \sqrt{N_{total}}$ is the optimal entropy of the system; $-\sum_{j=1}^m p_j \log_2 p_j$ is the current entropy of the system.

The relative distance between the feature vector ε_t of the generated scenario and the features of the original scenario is adopted to measure the degree δ to which the generated scenario τ preserves the characteristics of the initial scenario. Its formula is expressed as follows:

$$\delta = \frac{\sum_{l=1}^L |\tau_t^i - Es(t)|}{\sum_{l=1}^L |Es(t)|} \times 100\% \quad (6)$$

Here, L is the dimension of the scenario parameter vector ε ; $Es(t)$ ($t = 1, 2, \dots, T$) is the scenario composed of the expected values of historical scenarios in time T .

As systematically depicted in Fig.2, base on tasks, domain data and knowledge, the scenario generator framework operates through three LLM-powered agents in a closed-loop architecture: The Planner Agent (PA) initiates the process by synthesizing system constraints through domain knowledge integration, followed by the Environment Agent (EA) generating extreme scenarios with long-tail distributions via adversarial semantic modeling. Concurrently, the Social Agent (SA) constructs heterogeneous agent populations by extracting social network backbones and behavioral strategy library [27]. These components collectively formulate multi-temporal system states ε_t , which feed into iterative experimental calibration through a feedback mechanism, ultimately enabling self-optimizing generation of scenario sets $S = \{\tau^1, \tau^2, \dots, \tau^n\}$ for adaptive service ecosystem experimentation.

B. The Environment Generation by EA

EA aims to calculate the upper and lower limit boundaries of the environmental distribution based on the historical dataset, domain knowledge base, and adversarial prompt set. The detail is shown in Algorithm 1. The innovation of EA is mainly reflected in two aspects.

1) In the scenario generation stage, by integrating the semantic understanding ability of LLM, the professional knowledge of the domain knowledge base, and adversarial prompts, it breaks through the limitation of traditional methods that

only rely on the statistical features of historical data. It can generate more diverse and realistic scenarios, especially extreme scenarios with long-tail distributions.

2) The scenario adversarial verification mechanism within the reinforcement learning framework is introduced to evaluate and screen the credibility of the generated scenarios, ensuring that the high-risk scenarios finally used for calculating the boundaries have high authenticity and reliability. This improves the accuracy and effectiveness of calculating the environmental distribution boundaries, providing a more scientific and comprehensive basis for demand environment modeling.

Algorithm 1: Demand Environment Modeling and Extreme Environment Generation Algorithm by EA

Input: Historical dataset D , Domain knowledge base K , Adversarial prompt set P

Output: Upper and lower limit boundaries of the environmental distribution, V_{max} , V_{min}

- 1 Initialize $C \leftarrow \emptyset$, $S \leftarrow \emptyset$, $R \leftarrow \emptyset$, $C_{high_risk} \leftarrow \emptyset$;
- 2 **for** each d in D **do**
 - 3 Extract semantic features c from d using LLM and add c to C ;
 - 4 Extract event logic rule set L from K via knowledge distillation;
 - 5 **for** each p in P **do**
 - 6 Initialize $S_p \leftarrow \emptyset$;
 - 7 **for** each c in C **do**
 - 8 **for** each l in L **do**
 - 9 Generate scenario s by combining c , l , and p using LLM and add s to S_p ;
 - 10 $S \leftarrow S \cup S_p$;
 - 11 **for** each s in S **do**
 - 12 Set up an adversarial RL environment with s as input, calculate credibility score r_s and add it to R ;
 - 13 Set thresholds θ_{high} and θ_{risk} ;
 - 14 **for** $i \leftarrow 1$ to $|S|$ **do**
 - 15 **if** $R[i] \geq \theta_{high}$ **then**
 - 16 Calculate risk value $risk_i$ for $S[i]$;
 - 17 **if** $risk_i \geq \theta_{risk}$ **then**
 - 18 $S_{high_risk} \leftarrow S_{high_risk} \cup \{S[i]\}$;
 - 19 Extract data vector set V from relevant variables in S_{high_risk} ;
 - 20 Calculate lower and upper limit boundaries of environmental distribution from min and max values of variables in V ;
 - 21 **return** The upper and lower limit boundaries of the environmental distribution as V_{max} , V_{min} ;

C. The Social Collaboration Generation by SA

SA takes the social system data D as input and aims to generate the backbone network N of the relational structure. The detail is shown in Algorithm 2. Its innovations are reflected in two aspects.

1) During the feature extraction phase, by leveraging the powerful semantic understanding and analysis capabilities of LLMs, it can deeply explore the latent characteristics of individuals and groups. Compared with traditional methods, it can obtain more abundant and accurate information, providing a solid foundation for subsequent relational network construction and behavior strategy generation.

2) In the process of relational network construction, it comprehensively considers the relationships among individuals, between individuals and groups, and among groups. By setting reasonable relationship strength thresholds, it accurately screens out meaningful relationships, making the constructed relational network more consistent with the complex structure of the actual social system.

Algorithm 2: Social Relationship Structure Generation Algorithm by SA

Input: Social system data D

Output: Skeleton of the relationship network N

- 1 Initialize $I \leftarrow \emptyset$, $G \leftarrow \emptyset$, $I_{features} \leftarrow \emptyset$, $G_{features} \leftarrow \emptyset$, $N \leftarrow \emptyset$;
- 2 **for** each r in D **do**
 - 3 **if** r is individual **then**
 - 4 Add r to I ;
 - 5 **else if** r is group **then**
 - 6 Add r to G ;
- 7 **for** each i in I **do**
 - 8 Add LLM(i) to $I_{features}$;
- 9 **for** each g in G **do**
 - 10 Add LLM(g) to $G_{features}$;
- 11 **for** each i in I **do**
 - 12 **for** each j in I where $i \neq j$ **do**
 - 13 Calculate s_{ij} based on features of i and j ; **if** s_{ij} meets threshold M_{ij} **then**
 - 14 Add (i, j, s_{ij}) to N ;
- 15 **for** each g in G **do**
 - 16 Calculate s_{ig} based on features of i and g ; **if** s_{ig} meets threshold M_{ig} **then**
 - 17 Add (i, g, s_{ig}) to N ;
- 18 **for** each g_1 in G **do**
 - 19 **for** each g_2 in G where $g_1 \neq g_2$ **do**
 - 20 Calculate $s_{g_1 g_2}$ based on features of g_1 and g_2 ;
 - 21 **if** $s_{g_1 g_2}$ meets threshold $M_{g_1 g_2}$ **then**
 - 22 Add $(g_1, g_2, s_{g_1 g_2})$ to N ;

D. The Rule Generation and Scheme Calibration by PA

PA takes industry specification documents I and expert experience E as inputs, aiming to generate executable logical expressions L and an optimized experiment scheme P . The detail is shown in Algorithm 3. The innovations of PA are mainly reflected in two aspects.

1) During the generation of system constraint conditions, by integrating the Retrieval-Augmented Generation (RAG) technology with LLM [28], PA combines industry specification documents and expert experience. This approach enables the full exploration and utilization of domain knowledge, resulting in system constraint conditions that are more in line with real-world scenarios. Compared with traditional methods, the generated constraint conditions are more comprehensive and accurate.

2) In the processes of rule compilation and experiment plan optimization, PA employs Program-aided Prompting technology to automatically compile natural language rules into executable logical expressions and achieves self-calibrating optimization of the experiment plan. This not only improves the efficiency and accuracy of rule compilation but also allows for dynamic adjustment of the experiment plan based on experimental results. Consequently, the finally generated executable logical expressions and experiment plan can better meet practical needs, providing a more intelligent and efficient method for the generation of system rules.

IV. EXPERIMENTAL DESIGN AND RESULT ANALYSIS OF COMPUTATIONAL EXPERIMENTS

Compared with the current transaction-based revenue-sharing governance strategy adopted by the World's Largest Public API Marketplace, "RapidAPI", as of March 2025, the number of developers has reached over 4 million, the number of APIs has exceeded 40,000. The ProgrammableWeb ecosystem governance adopted a non-direct revenue strategy and was finally forced to be decommissioned, which highlights the importance of service ecosystem governance. Many researchers intend to conduct in-depth studies on its governance approach. However, it is difficult to formulate experimental plans and the experimental efficiency is low. Based on such requirements, we carry out extensive experiments to answer the following research questions regarding the scenario generator design method proposed in this paper:

- **RQ1:** Can it take into account the extreme scenarios in the governance of the service ecosystem?
- **RQ2:** Can it effectively represent the collaborative characteristics of individuals within the service ecosystem?
- **RQ3:** Are the generated scenarios more accurate and efficient?

A. Experimental Design

ProgrammableWeb is an authoritative online platform that focuses on the API (Application Programming Interface) ecosystem. It was established in 2005. In the 2010s, with the explosion of the API economy (such as the open APIs of Twitter, Google Maps, etc.), ProgrammableWeb became an

Algorithm 3: Rule Generation Algorithm by PA

Input: Industry specification documents I , Upper and lower limit boundaries of the environmental distribution V_{max} and V_{min} , Set of key relationships for each year N (including initial relationships N_0)

Output: Executable logical expressions L , Optimized experiment scheme P

```

1 Initialize system constraint set  $C \leftarrow \emptyset$ ;
2 for each document  $d$  in  $I$  do
3   |   constraint  $\leftarrow$  RAG( $d$ , LLM);
4   |   Add constraint to the system constraint set  $C$ ;
5 Initialize executable logical expression set  $L \leftarrow \emptyset$ ,
   Initialize the environmental scheme  $Ev^{(0)}$ ;
6 while not converged do
7   |   Compute gradient:  $\nabla J = (\frac{\partial J}{\partial Ev}, \frac{\partial J}{\partial N})$  where
         $J = J(Ev, N)$ ;
8   |   Update parameters:  $Ev^{(k+1)} \leftarrow Ev^{(k)} - \eta \cdot \nabla J$ ;
9   |   if  $\|\nabla J\| < \epsilon$  ( $\epsilon$  is the preset convergence threshold) then
10    |     |   Break
11 for each constraint  $c$  in  $C$  do
12   |   expression  $\leftarrow$  Program_aided Prompting( $c$ , LLM);
13   |   Add expression to the executable logical
        expression set  $L$ ;
14 Initialize experiment scheme  $P \leftarrow$  InitialScheme( $L$ );
15 while not converged do
16   |   results  $\leftarrow$  Execute( $P$ );
17   |   feedback  $\leftarrow$  Analyze(results);
18   |    $P \leftarrow$  Adjust( $P$ , feedback,  $L$ );
19   |   if isConverged(feedback) then
20    |     |   converged  $\leftarrow$  true;
21 return  $L, P$ ;

```

industry benchmark. As market competition intensified (e.g., RapidAPI was founded in 2014), it faced the problems of user loss and declining revenue, making it difficult to cover the maintenance costs. As a result, it was deactivated in 2022. Based on the ProgrammableWeb dataset, we designed experiments to analyze the evolution process of the ProgrammableWeb service ecosystem within its life cycle under a fluctuating social environment.

As shown in Fig.3, based on the analysis of the ProgrammableWeb dataset, social characteristics of environments and entities were extracted to form an initial distribution from raw data. Different experimental schemes were generated under distinct experimental design methods. Finally, quality and efficiency comparisons were conducted for scenarios generated by different schemes. In addition to **Method 1 (Original)** using raw data statistical analysis, we classified current research techniques in service ecosystem governance into four experimental design methods based on the summary

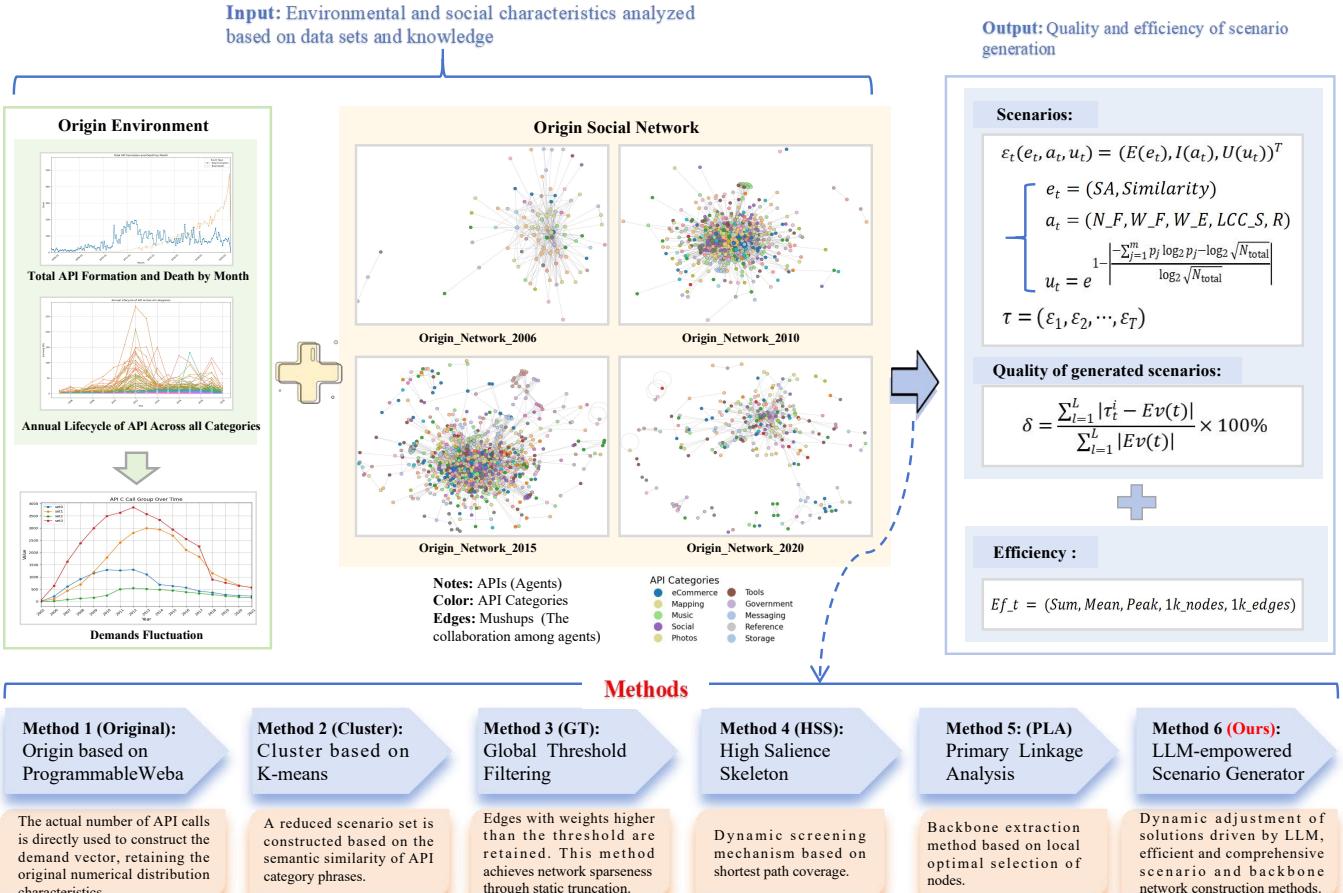


Fig. 3. The framework of computational experiments design. Using the analysis of environmental and social characteristics based on datasets and knowledge as input, six different experimental methods are employed for processing, and a scenario set along with evaluation indicators are outputted, including the assessment of the quality of generated scenarios and experimental efficiency.

of complex network Analysis and ABM in Section II:

Experimental Methods:

- Method 2: Cluster.** The K-means algorithm is used to partition the initial distribution into clusters. The cosine similarity between samples and cluster centers is calculated, and adaptive thresholding is then used to remove outliers and noise, producing the final experimental plan.
- Method 3: Global Threshold Filtering (GT).** Through static truncation, edges with weights exceeding a pre-defined threshold are retained, enabling sparsification of samples under given environmental states and narrowing the distribution of generated scenarios.
- Method 4: High Salience Skeleton (HSS).** Significant service dependency relationships are preserved based on statistical hypothesis testing. A single-source Dijkstra's algorithm is applied to generate a shortest path tree for each node. The significance score of each edge is calculated according to path coverage frequency, and edges with scores above the threshold are retained.
- Method 5: Primary Linkage Analysis (PLA).** For each element or node, the adjacent edge with the maximum weight is selected, and only bidirectional maximum

weight connections are retained to form an undirected network.

Scenario Vector: Based on Formula (1), the scenario vector during the experiment is represented as follows:

$$e_t = (SA, \text{Similarity}) \quad (7)$$

Here, *SA* (i.e., *SumActivity*) is the total activity volume, which is the sum of times count APIs are called by Mashup, reflecting demand. APIs are divided into four categories by property. *Similarity* monitors changes in market demand structure for each year according to the proportional distribution of the count APIs are called by Mashup.

$$a_t = (NF, WF, WE, LCC_S, Reachability) \quad (8)$$

Here, *NF* (i.e., *NodeFraction*) is the sub-graph to original graph node ratio, reflecting node retention integrity. *WF* (i.e., *WeightFraction*) is the ratio of retained edge to original network total weight, indicating core connection retention degree. *WE* (i.e., *WeightEntropy*) is the edge weight probability distribution, measuring weight distribution diversity. *LCC_S* (i.e., *LCCSize*) is the largest connected component's node number, measuring network structure integrity. *Reachability*

is the proportion of node pairs with at least one path, showing network global connectivity and structural compactness. A value nearer 1 means a more compact network.

For API nodes of each year, logarithmic standardization is performed by comprehensively considering the characteristics of market demand and network structure. The total number of APIs is N_{total} , and the formula for calculating the individual effectiveness I_U_i based on Formula (4) is as follows:

$$I_U_i = \alpha \cdot \ln \left(\frac{C_i}{C_{\max}} + 1 \right) + \beta \cdot \ln \left(\frac{D_i}{D_{\max}} + 1 \right) \quad (9)$$

Here, C_i is the total number of times the current API node is called by Mashup in the current year. C_{\max} is the maximum number of times an API node is called by Mashup among all API nodes in the current year. D_i is the network weighted degree centrality, with the formula $\sum_{j \in N(i)} w_{ij}$, where w_{ij} represents the edge weight. D_{\max} is the maximum weighted degree in the network in the current year. α is the weight coefficient for market demand, adjusting its contribution. β is the weight coefficient for network influence, regulating the contribution of the collaborative network.

Sort the individual utility values in descending order and use the equal-width binning method to define m niches ($m \approx \sqrt{N_{total}}$) [26]. Based on Formula (4), the system effectiveness u_t during time period t is derived.

Evaluation Indicators: The quality of generated scenarios is obtained based on Formula (1) and (6). The experimental efficiency is represented by the following indicators.

- *Sum(ks)*: Total time of multiple experiments, measuring resource consumption in long-term or batch tasks.
- *Mean(ks)*: Average time per single experiment, reflecting processing efficiency in regular scenarios.
- *Peak(ks)*: Maximum single-experiment time during the process, evaluating method stability in the worst - case.
- *Time per 1k nodes (/1kn) and per 1k edges (/1ke)*: Quantify the sensitivity of different methods to node and interaction scale. Lower values imply stronger large-scale network processing.

B. Experimental Analysis

Each experimental design method yields a final scheme. Executing the final scheme generates a corresponding scenario set. We randomly select one scenario for in - depth analysis to sequentially answer the three research questions.

1) *Analysis of the Generated Requirement Environment (RQ1):* We collected the number of active APIs on ProgrammableWeb and RapidAPI from 2005 to 2022. The discrete API categories (such as eCommerce, Social) were mapped to a continuous semantic space using a pre-trained BERT model. Functionally similar categories were merged by an LLM agent, ultimately forming four major semantic communities. Finally, APIs were classified into four categories: Infrastructure, Lifestyle Services, Business Management, and Social Entertainment. For each of these four API sets, the number of times they were called by Mashups was counted

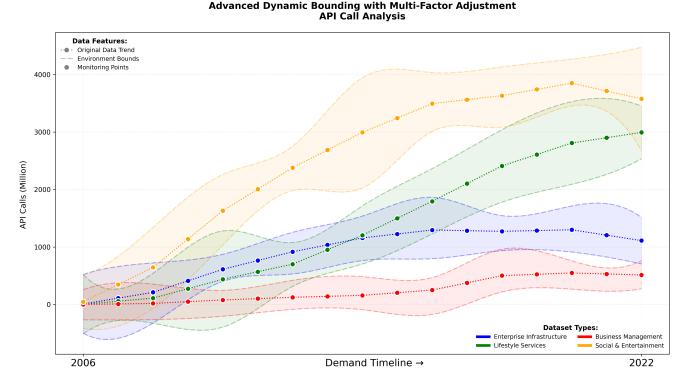


Fig. 4. Results of Extreme Environment Generation

to construct a demand time series, which was used to analyze changes in external market environment.

Based on this, the environment agent in the scenario generator fits the evolution trend of market demand. The LLM flexibly represents composite patterns such as growth, saturation, and oscillation by collecting online information and making inferences. By analyzing various data, the impact weights at different time scales are quantified. As shown in Fig.4, our method determines the extreme boundary of environmental fluctuations at different times, and the environmental coverage has been significantly improved.

2) *Analysis of the Generated Social collaboration (RQ2):* Social interaction characteristics within the ProgrammableWeb ecosystem are represented by an API-API co-occurrence network. Here, Mashup serves as the service composition unit, and API as the basic functional module. Weighted edges are constructed by counting the co-occurrence frequencies of APIs within the same Mashup, generating co-occurrence network graphs for the years 2005-2020. In the original state (Original), there are numerous API nodes with dense connections and complex co-occurrence relationships. The corresponding API types are diverse, conforming to the complexity of a real-world service ecosystem network. Based on the six methods above, an API-API co-occurrence network graph for each year is drawn. Nodes are colored according to the community division results of each network, as shown in Fig.5.

To visually verify the ability of different methods to represent the core structure of the network, this paper selects the social association state diagrams of the initial state (2006), intermediate states (2010 and 2015), and the later state (2020) for comparative analysis from visual dimensions such as community consistency, key node retention, and color distribution fidelity. By analyzing the network evolution diagrams, it can be seen that the Cluster method disrupts the topology-demand coupling due to rigid semantic screening, leading to the simplification of community structure. GT and HSS, limited by static thresholds or path redundancy, sacrifice key connections for cross-category collaboration. The tree-like constraint of PLA causes over-screening and simplification, with the number of communities and color distribution deviating significantly

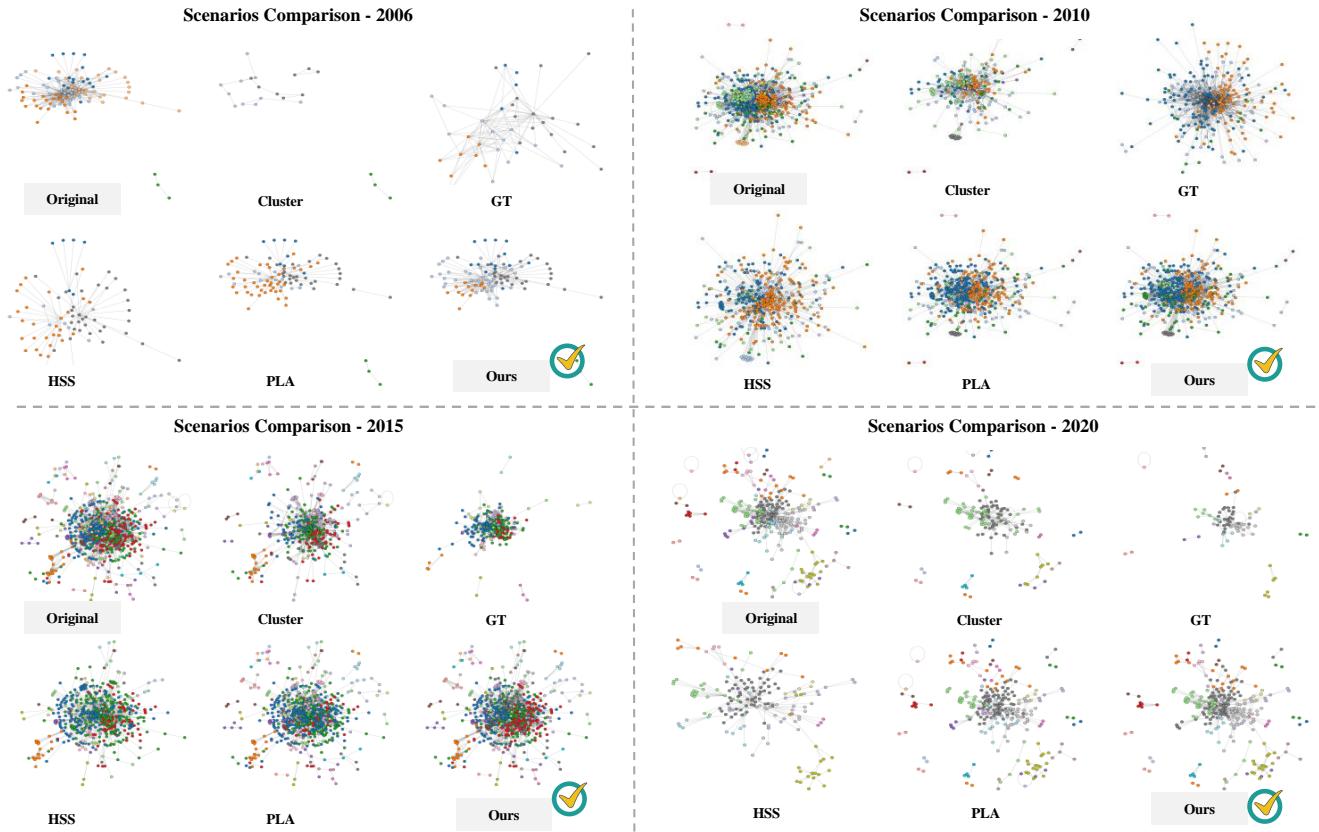


Fig. 5. The results of social collaborative relationship generation. It shows the comparison results generated by six methods, including Original, Cluster, GT, HSS, PLA and Ours, throughout the life course of ProgrammableWeb's early (2006), mid-term (2010 and 2015) and late (2020) development. Among these methods, our method is most consistent with the original relational structure.

TABLE I
STATISTICAL DATA ON 8-DIMENSIONAL SCENARIO VECTOR INDEXES AND TOTAL DEVIATION OF EACH METHOD

| Method | SA | Similarity | NF | WF | WE | LCC_S | Reachability | VE (u) | Deviation(δ) |
|----------|---------|------------|--------|--------|--------|--------|--------------|--------|-----------------------|
| Original | 0.9776 | 1 | 1 | 1 | 1 | 1 | 0.9712 | 0.8608 | - |
| Cluster | 0.3587 | 0.622 | 0.4643 | 0.226 | 0.8272 | 0.418 | 0.7886 | 0.9865 | 0.4329 |
| GT | 0.8578 | 0.8179 | 0.3942 | 0.5397 | 0.7869 | 0.3894 | 0.9427 | 0.9093 | 0.2972 |
| HSS | 0.92287 | 0.7753 | 0.7816 | 0.2214 | 0.7009 | 0.4214 | 0.3885 | 0.8114 | 0.3556 |
| PLA | 0.9448 | 0.7987 | 1 | 0.2997 | 0.7087 | 0.674 | 0.4826 | 0.8049 | 0.2685 |
| Ours | 0.9817 | 0.9767 | 0.9978 | 0.6335 | 0.8973 | 1 | 0.9755 | 0.824 | 0.0783 |

TABLE II
STATISTICS ON EXPERIMENTAL EFFICIENCY TIMES OF DIFFERENT METHODS

| Method | Sum(ks) | Mean(ks) | Peak(ks) | /lkn(ks) | /lke(ks) |
|----------|--------------|--------------|--------------|--------------|--------------|
| Original | 13.037 | 0.869 | 10.142 | 25.775 | 4.166 |
| Cluster | 18.977 | 1.265 | 18.142 | 76.665 | 22.430 |
| GT | 6.181 | 0.412 | 5.671 | 31.862 | 8.813 |
| HSS | 5.709 | 0.381 | 5.314 | 14.329 | 14.884 |
| PLA | 5.937 | 0.396 | 5.393 | 11.739 | 12.701 |
| Ours | 3.171 | 0.211 | 1.585 | 6.279 | 1.907 |

from the benchmark network, poorly reflecting the mesh-like interactions in the real ecosystem. In contrast, Ours maintains a

community structure and color distribution basically consistent with Original in all four stages, retains cross-community connections, and the color-mixing pattern is highly consistent with Original, being able to reflect the real-world collaboration pattern.

3) *Analysis of the Quality and Efficiency of Scenario Generator (RQ3):* This paper comprehensively assesses the differences between the experimental design method of our proposed scenario generator and other methods through multi-dimensional analysis.

Performance Evaluation. Based on the calculation and comparison of the characteristic parameters of the 8-dimensional scenario vector, as shown in Fig.6, taking the

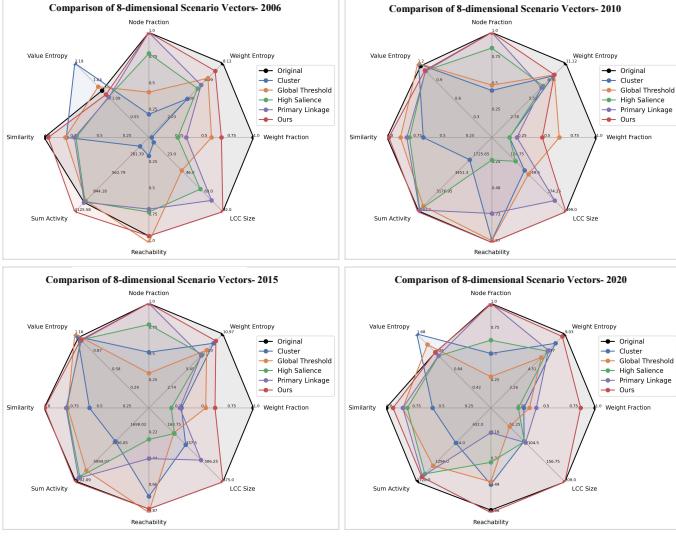


Fig. 6. The effect of scenario generator. It presents the comparison of 8-dimensional scenario vectors for the scenarios generated by six methods: Original, Cluster, GT, HSS, PLA, and Ours. Among them, our method aligns most closely with the comprehensive features of the original scenario.

Original method as the benchmark to compare the scenario states in multiple time periods, the scenarios generated by our method show the best overall agreement with those of the Original method. In Table I, the comprehensive difference degree (δ) of our method in eight core indicators is only 0.0738, which represents a 72.5% improvement in accuracy compared to the sub-optimal method PLA with a value of 0.2685. In the environmental dimension (i.e., SA, Similarity), the environmental features generated by our method are close to those of the Original method, with improvements of 14.5% and 19.4% respectively compared to the sub-optimal method GT. Moreover, other methods suffer from issues such as the loss of key APIs, sacrifice of long - tail requirements, or structural deviation. Regarding social relationship generation (i.e., NF, WF, WE, LCC_S, Reachability), our method is close to the original network in terms of network integrity, global reachability, and weight distribution diversity. In contrast, methods like GT, Cluster, HSS, and PLA exhibit varying degrees of adaptability defects or connectivity losses. In terms of system health (i.e., VE), the deviation of our method from the Original network is significantly better than other methods.

Experimental Efficiency. Calculated based on Table II, the total time for multiple rounds of experiments of our method is 3.171 thousand seconds. This is 75.7% lower than the 13.037 thousand seconds of the benchmark Original method and 44.5% more optimized than the 5.709 thousand seconds of the sub-optimal method HSS. Our method outperforms in core indicators such as time per thousand nodes, time per thousand edges, and peak time. It also demonstrates excellent stability, with the average time being significantly better than that of the comparison methods. Additionally, the growth rate of the time per thousand nodes/edges of our method is lower than the network scale growth rate, indicating its potential to handle

large-scale data and providing methodological support for the research on the governance of service ecosystems.

In conclusion, through dynamic weight adjustment and spatio-temporal coupling modeling, our method effectively resolves the performance contradictions of existing methods. It achieves coordinated optimization in environmental distribution, integrity of social relationship structure, and improved experimental efficiency greatly, thus providing a new paradigm for conducting experiments in the research on the governance of service ecosystems.

V. CONCLUSION

In summary, this study proposes an LLM-empowered agents collaboration framework for generating governance-referenced scenarios in dynamic service ecosystems. The approach coordinates three specialized agents: the Environment Agent (EA) simulates extreme social environments, the Social Agent (SA) generates features and collaborative networks, and the Planner Agent (PA) couples task-role relationships and optimizes task solutions. Through autonomous experimental regulation, the framework quantifies qualitative features and produces high-quality, computationally efficient scenario generation schemes. Experiments on the ProgrammableWeb dataset demonstrate that the method generates more accurate scenarios, offering a novel and effective approach for constructing experimental systems tailored to service ecosystem governance.

Despite its strengths, the current framework has limitations, including potential semantic biases inherent in LLMs and scalability constraints when handling large-scale, complex scenarios. Future research should focus on integrating reinforcement learning for adaptive exploration, developing lightweight verification modules to enhance reliability, and incorporating causal frameworks to better address intricate social interdependencies. These advancements will further strengthen the framework, providing robust governance tools for evolving service ecosystems and enabling more effective decision-making in dynamic social environments.

ACKNOWLEDGMENT

This work has been supported in part by the Natural Science Foundation of China (No.92367202), National Natural Science Foundation of China (No.62472306, No.62441221, No.62206116), Tianjin University's 2024 Special Project on Disciplinary Development (No.XKJS-2024-5-9), Tianjin University Talent Innovation Reward Program for Literature & Science Graduate Student (C1-2022-010), and Henan Province Key Research and Development Program (No.251111210500), Shandong Taishan Industry Leading Talent Project (NO.tsrx202211010).

REFERENCES

- [1] Z. Wu, J. Yin, S. Deng, J. Wu, Y. Li, and L. Chen, "Modern service industry and crossover services: Development and trends in china," *IEEE Transactions on Services Computing*, vol. 9, no. 5, pp. 664–671, 2016.
- [2] X. Xue, X. Yu, D. Zhou, C. Peng, X. Wang, D. Liu, and F.-Y. Wang, "Computational experiments for complex social systems—part iii: the docking of domain models," *IEEE Transactions on Computational Social Systems*, vol. 11, no. 2, pp. 1766–1780, 2023.

- [3] X. Xue, G. Li, D. Zhou, Y. Zhang, L. Zhang, Y. Zhao, Z. Feng, L. Cui, Z. Zhou, X. Sun *et al.*, “Research roadmap of service ecosystems: A crowd intelligence perspective,” *International Journal of Crowd Science*, vol. 6, no. 4, pp. 195–222, 2022.
- [4] X. Xue, D. Zhou, X. Yu, G. Wang, J. Li, X. Xie, L. Cui, and F.-Y. Wang, “Computational experiments for complex social systems: Experiment design and generative explanation,” *IEEE/CAA Journal of Automatica Sinica*, vol. 11, no. 4, pp. 1022–1038, 2024.
- [5] M. Lu, S. Chen, X. Xue, X. Wang, Y. Zhang, Y. Zhang, and F.-Y. Wang, “Computational experiments for complex social systems—part ii: The evaluation of computational models,” *IEEE Transactions on Computational Social Systems*, vol. 9, no. 4, pp. 1224–1236, 2021.
- [6] X. Xue, F. Chen, D. Zhou, X. Wang, M. Lu, and F.-Y. Wang, “Computational experiments for complex social systems—part i: The customization of computational model,” *IEEE Transactions on Computational Social Systems*, vol. 9, no. 5, pp. 1330–1344, 2021.
- [7] X. Xue, X. Yu, D. Zhou, X. Wang, C. Bi, S. Wang, and F.-Y. Wang, “Computational experiments for complex social systems: Integrated design of experiment system,” *IEEE/CAA Journal of Automatica Sinica*, vol. 11, no. 5, pp. 1175–1189, 2024.
- [8] H. Liu, X. Huang, M. Jia, T. Jia, J. Han, Y. Li, and Z. Wu, “Uacad: Unsupervised adversarial contrastive learning for anomaly detection on multi-modal data in microservice systems,” *IEEE Transactions on Services Computing*, 2024.
- [9] M. Liu, Z. Tu, X. Xu, Z. Wang, and Y. Wang, “A data-driven approach for constructing multilayer network-based service ecosystem models,” *Software and Systems Modeling*, vol. 22, no. 3, pp. 919–939, 2023.
- [10] M. M. Shin, S. Jung, and J. S. Rha, “Study on business ecosystem research trend using network text analysis,” *Sustainability*, vol. 13, no. 19, p. 10727, 2021.
- [11] M. Liu, Z. Tu, H. Xu, X. Xu, and Z. Wang, “Community-based service ecosystem evolution analysis,” *Service Oriented Computing and Applications*, vol. 16, no. 2, pp. 97–110, 2022.
- [12] K. Huang, Y. Fan, and W. Tan, “An empirical study of programmable web: A network analysis on a service-mashup system,” in *2012 IEEE 19th international conference on web services*. IEEE, 2012, pp. 552–559.
- [13] A. Yassin, A. Haidar, H. Cherifi, H. Seba, and O. Togni, “An evaluation tool for backbone extraction techniques in weighted complex networks,” *Scientific Reports*, vol. 13, no. 1, p. 17000, 2023.
- [14] C. Villalba and F. Zambonelli, “Towards nature-inspired pervasive service ecosystems: Concepts and simulation experiences.” *Journal of Network and Computer Applications*, vol. 34, no. 2, pp. 589–602, 2011.
- [15] D. Ye, Q. He, Y. Wang, and Y. Yang, “An agent-based integrated self-evolving service composition approach in networked environments,” *IEEE Transactions on Services Computing*, vol. 12, no. 6, pp. 880–895, 2016.
- [16] S. L. Lim and P. J. Bentley, “How to be a successful app developer: Lessons from the simulation of an app ecosystem,” *Acm Sigevolution*, vol. 6, no. 1, pp. 2–15, 2012.
- [17] X. Xue, S. Wang, L. Zhang, Z. Feng, and Y. Guo, “Social learning evolution (sle): Computational experiment-based modeling framework of social manufacturing,” *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3343–3355, 2018.
- [18] D. Zhou, X. Xue, and Z. Zhou, “Sle2: The improved social learning evolution model of cloud manufacturing service ecosystem,” *IEEE Transactions on Industrial Informatics*, vol. 18, no. 12, pp. 9017–9026, 2022.
- [19] X. Xiao, Y. Xiang-Ning, Z. De-Yu, P. Chao, W. Xiao, Z. Zhang-Bing, and W. Fei-Yue, “Computational experiments: Past, present and perspective,” *Acta Automatica Sinica*, vol. 49, no. 2, pp. 246–271, 2023.
- [20] X. Xue, X. Yu, and F.-Y. Wang, “Chatgpt chats on computational experiments: From interactive intelligence to imaginative intelligence for design of artificial societies and optimization of foundational models,” *IEEE/CAA Journal of Automatica Sinica*, vol. 10, no. 6, pp. 1357–1360, 2023.
- [21] H. Jung, W. Seo, S. Song, and S. Na, “Toward value scenario generation through large language models,” in *Companion Publication of the 2023 Conference on Computer Supported Cooperative Work and Social Computing*, 2023, pp. 212–220.
- [22] C. W. Davis, A. J. Jetter, and P. J. Giabbanelli, “Towards an automatic construction of simulation scenarios: A systematic review,” in *2023 Winter Simulation Conference (WSC)*. IEEE, 2023, pp. 2494–2505.
- [23] C. Lu, “Test scenario generation for autonomous driving systems with reinforcement learning,” in *2023 IEEE/ACM 45th International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)*. IEEE, 2023, pp. 317–319.
- [24] X. Xue, Y. Shen, X. Yu, D.-Y. Zhou, X. Wang, G. Wang, and F.-Y. Wang, “Computational experiments: A new analysis method for cyber-physical-social systems,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 54, no. 2, pp. 813–826, 2023.
- [25] J. Li, J. Zhou, and B. Chen, “Review of wind power scenario generation methods for optimal operation of renewable energy systems,” *Applied Energy*, vol. 280, p. 115992, 2020.
- [26] X. Yu, X. Xue, D. Zhou, L. Fang, and Z. Feng, “Beyond traditional metrics: The power of value entropy in multidimensional evaluation of the service ecosystem,” in *2024 IEEE International Conference on Web Services (ICWS)*. IEEE, 2024, pp. 611–621.
- [27] D. Zhou, X. Xue, X. Lu, Y. Guo, P. Ji, H. Lv, W. He, Y. Xu, Q. Li, and L. Cui, “A hierarchical model for complex adaptive system: From adaptive agent to ai society,” *ACM Transactions on Autonomous and Adaptive Systems*, 2024.
- [28] D. Zhou, X. Xue, Q. Ma, C. Guo, L. Cui, Y. Tian, J. Yang, and F. Wang, “Federated experiments: Generative causal inference powered by llm-based agents simulation and rag-based domain docking,” *IEEE/CAA Journal of Automatica Sinica*, vol. 11, no. 11, pp. 1–4, 2024.