

RESEARCH ARTICLE

Dynamic QoS Aware Service Composition Framework Based on AHP and Hierarchical Markov Decision Making

RUI WANG 

North China Institute of Aerospace Engineering, Langfang 065000, China

e-mail: wangrui@nciae.edu.cn


This work was supported by the School Level Project "Research on QoS Aware Service Composition in Cloud Computing Environment."

ABSTRACT In recent years, with the continuous development of service-oriented computing technology, the industry has increasingly high quality requirements for service solving. A dynamic perceptual service composition framework is proposed for solving Web service problems. During the process, the user provides three aspects of information, including user context, user requests, and user preferences. The relevant elements in the problem are divided into multiple logical levels and processed layer by layer. The entire service process is represented through a directed graph, and specific business is represented using a vertex set. A forward step re-planning rule is established to skip business vertices. The experimental results showed that the research method had an error of only 0.071% in solving quality analysis at 20 business vertices, which was lower compared with other methods. In the multi concurrency tolerance test, the research method only experienced 5 crashes during a 500s runtime. In the analysis of packet loss rate during operation, when there was no network fluctuation, the packet loss rate of the research method fluctuated between 0.1% and 0.7%. The research method can provide service discovery results that are more in line with the actual needs and preferences, which can provide better solution results for service composition problems.

INDEX TERMS Quality of service, analytical hierarchy process, Markov decision, service composition, directed graph, re-planning.

I. INTRODUCTION

In today's rapidly developing Internet era, the Internet model is changing from the traditional data-driven model to the service-oriented model [1]. The core of this transformation lies in services rather than simple data, which has become the core element of Internet architecture and application development. Since the emergence of computer software, the flexibility, scalability, correctness, and robustness of software systems have always been the core goal of software development [2]. Over time, these demands have not weakened, but have become increasingly strong,

The associate editor coordinating the review of this manuscript and approving it for publication was Liang-Bi Chen .

driving continuous innovation in software engineering practices. To effectively address these challenges, the software engineering encourages system construction based on existing software resources, rather than starting from scratch every time [3]. This component-based development paradigm greatly improves the efficiency and quality of software development, while also reducing development costs and time. The computation of services represents an emerging, component-based software development model [4]. In service computing, Quality of Service (QoS) describes the performance characteristics of Web services, which becomes the core part of their non-functional characteristics. When solving QoS related problems, Analytic Hierarchy Process (AHP) provides decision-makers with a method to handle

complex problems by establishing a hierarchical model, simplifying the decision-making process by decomposing the problem [5]. Markov decision, as an advanced decision-making model, is particularly suitable for dealing with decision-making problems in dynamic and uncertain environments. In the service composition framework, Markov decision-making can intelligently adjust service selection and composition strategies based on real-time QoS feedback and other environmental information. In this context, the study attempts to innovatively combine AHP and hierarchical Markov decision-making to establish a dynamic QoS aware service composition framework. Based on preferences and real-time QoS data, dynamic decisions are made to ensure that the service composition framework can flexibly respond to environmental changes while maintaining decision consistency to provide technical reference for the development of the computer industry.

The research is mainly conducted from four aspects. The first part discusses the research results related to QoS and AHP. The second part designs a dynamic QoS aware service composition framework based on AHP and hierarchical Markov decision-making. The third part analyzes the effectiveness of the research framework. The last part is a discussion and summary of the entire text.

II. RELATED WORKS

With the development of service-oriented computing, more scholars are beginning to realize the importance of QoS. Some scholars have conducted relevant research on QoS. Alaya B et al. proposed a technique based on interlayer methods to effectively maintain the QoS of video and audio streams. During the process, the downtime during video playback was eliminated, and the video data measured by roadside unit vehicles and nodes was maintained. The clock synchronization architecture was added for intelligent selection. The experimental results showed that the proposed method had a good video packet delivery rate [6]. Scholars such as Norouzi Shad M proposed a method based on intelligent decision support systems to address the quality of data propagation in QoS. During the process, location-based decision support was used to optimize the routing problem, and support vector machines and genetic algorithms were used to optimize clustering, ultimately making intelligent decisions. The experimental results indicated that the proposed method could effectively improve the network lifespan [7]. Beshley et al. proposed a method based on monitoring and agreement networks to ensure QoS according to user needs. The adaptive service priority sorting and server selected routing were used. The flow priority was changed using a software defined network controller combined with a service priority algorithm, and the QoS standard normalization value was integrated and added. The experimental results indicated that the proposed method had good user adaptability [8]. Costa et al. proposed a requirement analysis approach to

balance the complexity and accuracy of QoS. The resource allocation requirements for traffic prediction were analyzed, and the complexity of artificial intelligence algorithms was adjusted by utilizing contribution levels to adjust overall parameters. The experimental results indicated that the proposed method could effectively improve the quality of network services [9]. Scholars such as Li et al. proposed a trust aware approach to address the energy consumption and security issues of QoS routing protocols. The chaotic optimization strategy was used to initialize the ant colony population, and adaptive optimization strategy was used to dynamically adjust the algorithm trend, establishing a multi-objective chaotic elite adaptive algorithm. The experimental results showed that the proposed method could effectively reduce network energy consumption and improve service reliability [10].

Some scholars have conducted relevant research on AHP. Kaymaz et al. proposed an AHP-based method for evaluating sustainable development goals. The AHP analyzed different sustainable development goals, quantified the results of sustainable development, and used the quantitative results to output evaluation content. The experimental results indicated that the proposed method could effectively optimize the strategic content of sustainable development [11]. Senan et al. proposed an AHP-based method for assessing flood vulnerability in different regions of the West Gaozhi Mountains. The fuzzy strategy was introduced into the model, and vulnerability maps were divided through remote sensing satellite systems and geographic information systems. The map was validated using the working characteristic curves of the subjects. The experimental results indicated that the proposed method could provide support for flood control and disaster reduction strategies with higher accuracy [12]. Vilasan and Kapse combined AHP method with remote sensing and geographic information technology to evaluate the flood susceptibility of the Elnaculam region in Kerala. The results showed that the optimized model had a larger area under the working characteristic curve of the subjects, which was better than the traditional AHP in flood risk prediction. It could effectively reveal the main causes of floods [13]. To simplify the paired comparison burden in the decision-making process, Duleba proposed a multi-level minimalist AHP model and applied it at any decision level. The actual case test in Mersin, Türkiye confirmed the effectiveness of the multi-level simplified AHP model. The proposed method provided a new evaluation tool for the dynamic QoS aware service composition framework, which optimized service quality in changing network environments [14]. Mitra et al. successfully identified flood risk areas in the North Dinajipur region of India by combining geographic information technology and AHP, with 27.04% being medium risk areas, 15.62% being high-risk areas, and 4.59% being extremely high-risk areas. The model had good prediction accuracy, with a ROC-AUC value of 0.73. The study emphasized the importance of continuous improvement, which could

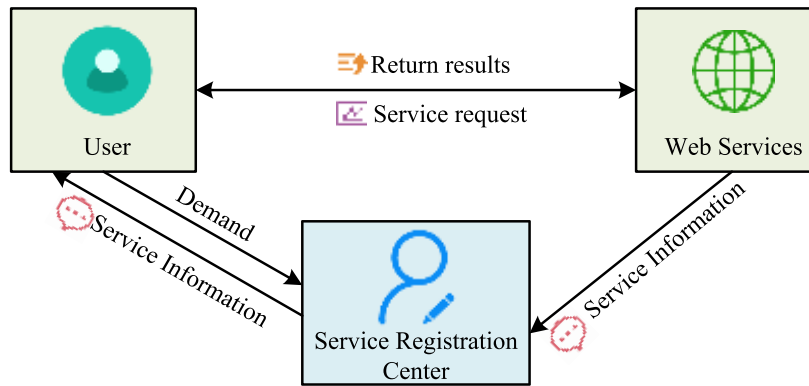


FIGURE 1. Classical SOA.

adapt to actual disaster management and provide insights for the study of dynamic QoS perception service composition frameworks [15].

In summary, although there have been many studies on QoS related guarantee methods, there is still a lack of methods that combine AHP. In view of this, the study attempts to conduct a dynamic QoS aware service composition framework based on AHP. It is expected to provide certain development references for information system technology. Compared with the QoS maintenance technology based on the inter-layer method in other existing technologies, the research method quantifies user preferences through AHP, which more accurately reflects the degree of users' emphasis on different QoS indicators. Compared with the QoS data propagation optimization method based on intelligent decision support system proposed by other advanced technologies, the research method can adjust the service selection and combination strategy in real time and adapt to dynamic and uncertain environment. However, the research method may be limited in dealing with the problems in the semantics-centric computing framework. The novelty of the research method lies in: (1) Combining AHP with hierarchical Markov decision process, it provides a new decision support tool for dynamic QoS sensing service composition. (2) Through the introduction of forward step replanning rules, the research method can flexibly skip the business apex in the process of service execution and effectively respond to environmental changes. The contributions of the research are as follows: (1) The research method provides a new service quality assurance mechanism for the field of service computing. (2) The research method has demonstrated its effectiveness and reliability in practical applications in solving quality analysis.

III. DESIGN OF HIERARCHICAL DYNAMIC QoS AWARE SERVICE COMPOSITION FRAMEWORK

A dynamic QoS aware service composition framework based on AHP and hierarchical Markov decision is designed. A QoS aware service based on user context is proposed in the Service Oriented Architecture (SOA) system, dividing the relevant elements in the problem into multiple logical levels, and

dividing the hierarchical Markov decision process into multiple sub processes and main processes.

A. QoS AWARE SERVICE GENERATION METHOD BASED ON AHP

QoS plays a crucial role in service computing as it directly relates to the efficiency, reliability, and user experience of services [16], [17]. In modern information systems and service computing environments, QoS is a key indicator for measuring service performance, ensuring that services can operate efficiently and stably while meeting user needs. In the SOA system, each service can have its own QoS requirements, which define the performance standards for service execution [18], [19]. SOA provides a framework that enables QoS management to be implemented at the service level. Based on the SOA, services can be classified, prioritized, and traffic controlled to ensure that critical services receive necessary resources and performance. The classical SOA system is shown in Figure 1.

In Figure 1, in the classical SOA model, there exists a triangular core relationship structure composed of Web services, service registry, and users. In the model, each Web service registers its detailed information to the service registry and applies to the service registry when users have service requirements. The service registration center then searches through all registered web services based on the specific needs of users, selects services that can meet these needs, and provides relevant information about these services to users [20], [21]. After receiving the information provided by the service registration center, users can directly interact with these Web services to achieve their business goals. The intermediate stage between service registration and usage is QoS aware service. A QoS aware service based on user context is proposed, as shown in Figure 2.

In Figure 2, in the QoS aware service based on user context, users provide three aspects of information including user context, user requests, and user preferences. User context refers to the user's environment and operational requests in a specific context, which can describe the user's specific needs and background. Service request refers to the specific

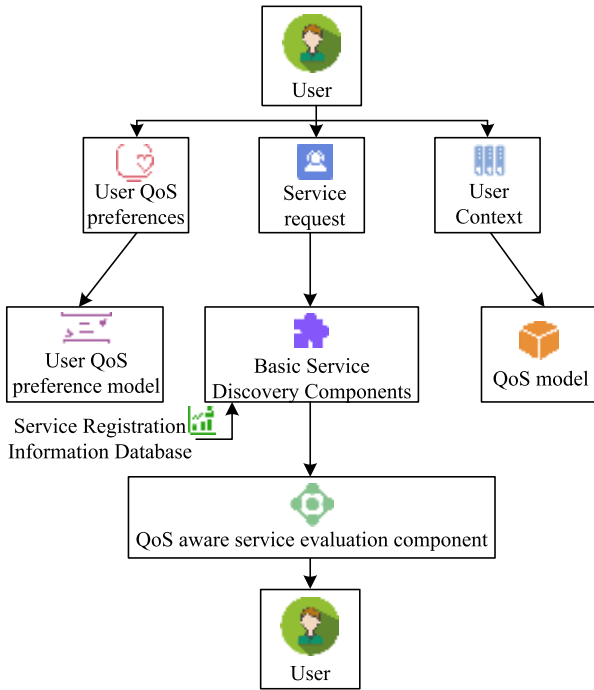


FIGURE 2. QoS aware service based on user context.

function that the user is seeking, describing the service content that the user wants to achieve, that is, the specific task that the user wants to perform. When implementing these services, commonly used technical solutions often include semantic techniques to ensure the accuracy of the services. Due to the fact that the QoS of Web services includes multiple different evaluation criteria, the personal preferences of users will have different impacts on these criteria. User preferences become a key factor in measuring the importance of each QoS standard to users [22], [23]. User context is processed through QoS models. Service requests are processed through basic service discovery components and combined with the service registration information library to obtain a collection of services. User preferences are processed through a user QoS preference model. The processing results are transmitted into the service evaluation component for sorting, and the optimal service is output to the user. When selecting the optimal service, it is necessary to comprehensively consider the impact of each QoS, which belongs to multi-dimensional decision-making problems. The study introduces AHP for service matching. User QoS preferences are set as a set of weights, where the condition for each weight value is shown in equation (1).

$$0 \leq w_i \leq 1, \sum_{i=1}^r w_i = 1 \quad (1)$$

In equation (1), w_i represents the weight of the i -th QoS indicator. r represents the number of weights. AHP is a qualitative method used to determine weights, which deduces the relative importance of various QoS indicators through a

simplified judgment process and allows for a certain degree of judgment inconsistency. The process involves constructing a comparison matrix of QoS indicators, where the entries reflect the comparison results between different indicators. The comparison matrix is shown in equation (2).

$$A = \begin{Bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nm} \end{Bmatrix} \quad (2)$$

In equation (2), A represents the comparison matrix. a represents the relative importance of two indicators. n represents the number of rows in the matrix. m represents the number of matrix columns. The relative importance of indicators in the comparison matrix needs to satisfy constraints, as shown in equation (3).

$$\begin{cases} a_{ii} = 1 \\ a_{ij} = 1/a_{ji} \\ a_{ij} = a_{ik}/a_{jk} \end{cases} \quad (3)$$

In equation (3), $a_{ij} = 1/a_{ji}$ represents the symmetry of importance between two indicators. The importance between two indicators is not related to the third indicator. If the maximum eigenvalue of the comparison matrix is larger than the number of rows, serious inconsistency will occur in the comparison matrix. The consistency of the matrix is judged by the consistency index, as shown in equation (4).

$$CI = \frac{\lambda_{Max} - n}{n - 1} \quad (4)$$

In equation (4), CI represents the consistency indicator. λ_{Max} represents the maximum eigenvalue of the matrix. After obtaining the preference ranking of QoS, AHP is used to refine the service evaluation and break down multiple QoS indicators for separate analysis and comparison. For each optional service, its performance is analyzed and discussed based on the corresponding QoS indicators. At the same time, the relevant elements in the problem are divided into multiple logical levels and processed them layer by layer. The service evaluation hierarchy is shown in Figure 3.

In Figure 3, in the service evaluation hierarchy, upper level elements exist in the form of relative goals as lower level elements, while lower level elements need to drive goal maximization. For any element in the upper layer, adjacent lower layer elements generate a comparison matrix through pairwise comparison. The comprehensive weight vector for each alternative service is shown in equation (5).

$$W_{Service} = W \times W_{QoS}^T \quad (5)$$

In equation (5), $W_{Service}$ represents the comprehensive weight of alternative services with the same number of rows as the comparison matrix. W_{QoS}^T represents the transpose of the QoS indicator weight vector. After solving for the weights, the final matching service is obtained as the final executed service.

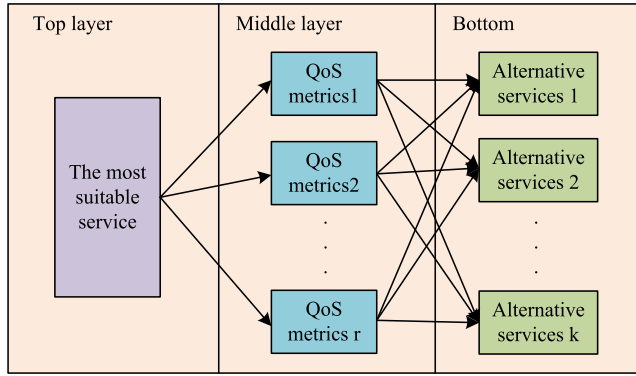


FIGURE 3. Service evaluation hierarchy.

B. QoS AWARE SERVICE SOLVING TECHNOLOGY COMBINING HIERARCHICAL MARKOV DECISION-MAKING

SOA provides consumers with the guarantee of dynamic identification and connection at runtime by implementing flexible connection methods between services [24], [25]. The core advantage of architecture lies in its ability to support flexible combinations of services, enabling multiple basic functional services to work together and achieve more complex business tasks, thereby creating additional value [26], [27]. A service that serves as the basic business function in a complete service process is called an atomic service. A set of atomic services, after being organized, forms a service process as a composite service [28], [29]. A directed graph is used to represent the entire service process, and a set of vertices is used to represent specific businesses. When combining services, each business vertex has a set of functionally equivalent atomic services that can be bound, represented by equation (6).

$$S_v = \{s_v^1, s_v^2, \dots, s_v^m\} \tag{6}$$

In equation (6), S_v represents the atomic service group. S_v^m represents atomic services. At the same time, each business vertex corresponds to a decision vector, as shown in equation (7).

$$D_v = (d_v^1, d_v^2, \dots, d_v^m) \tag{7}$$

In equation (7), D_v represents the decision vector. d_v^m represents the vector element. The vector element corresponding to the bound service is 1. Constraints in the form of service level agreements are used to constrain QoS metrics. The combination modes for atomic services include parallel mode, branch mode, pipeline mode, and loop mode are set. The combinatorial solution of atomic services belongs to the non-deterministic problem of polynomial complexity. Based on linear programming and local search techniques, a heuristic algorithm based on Local Branching is established to quickly solve the approximate optimal solution of the problem [30], [31]. The heuristic algorithm is based on the local branch principle and seeks the approximate optimal solution of the problem by step optimization. The first step of the

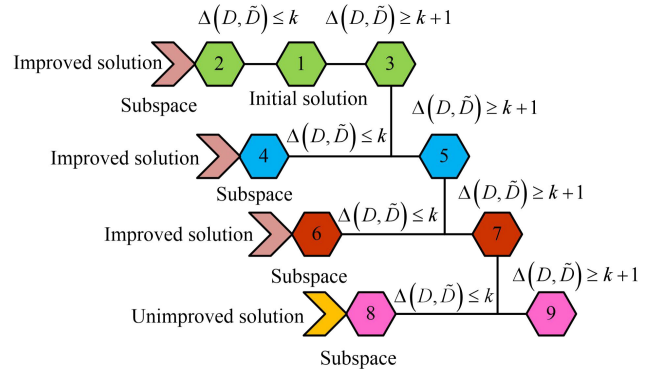


FIGURE 4. Heuristic algorithm search process based on local branching.

algorithm is to generate the initial solution, and then explore the solution space through local search. At each step, the algorithm evaluates the neighbors of the current solution and selects the one that makes the most improvement in the value of the objective function as the new current solution. This process is repeated until a stopping condition is met, such as reaching a preset number of iterations or no significant improvement in the quality of the solution. The decision variable constraints in the problem are relaxed and solved to obtain a relaxed linear programming. The linear programming solver is used to solve a relaxed linear programming problem, generate a preliminary solution, and define additional conditions for the problem, as shown in equation (8).

$$\Delta(D, \tilde{D}) = \sum_{d \in \beta} (1 - d) + \sum_{d \in \alpha \setminus \beta} d \leq k \tag{8}$$

In equation (8), $\Delta(D, \tilde{D})$ represents additional conditions. \tilde{D} represents the preliminary solution. k represents the given parameter of k-opt neighbors. The search process based on the heuristic algorithm of Local Branching is shown in Figure 4.

In Figure 4, during the search, the initial solution is used as the search root node, and the Local Branching constraint is used as the search branch point. The left branch of the search corresponds to the subspace of the given parameters with additional conditions less than or equal to k-opt neighbors. The right branch of the search corresponds to a subspace with an additional condition greater than or equal to the given parameter plus 1 for k-opt neighbors. The space is divided multiple times and recursively searched until the left branch no longer provides an improved solution. The additional ending condition is shown in equation (9).

$$node_{count} > node_{limit} \tag{9}$$

In equation (9), $node_{count}$ represents the number of search nodes. $node_{limit}$ represents the maximum number of nodes. When a set of atomic services is composed of multiple mixed patterns, it will form a complex service process. A mixed-structure service tree example is shown in Figure 5.

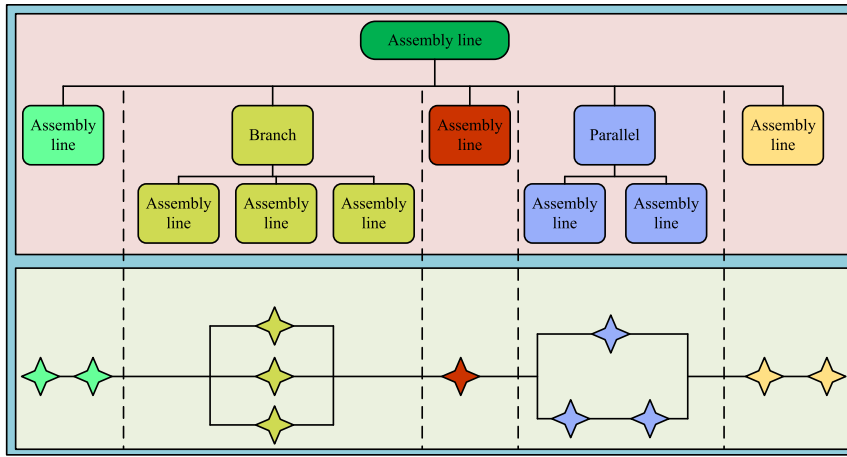


FIGURE 5. Example of a mixed-structure service tree.

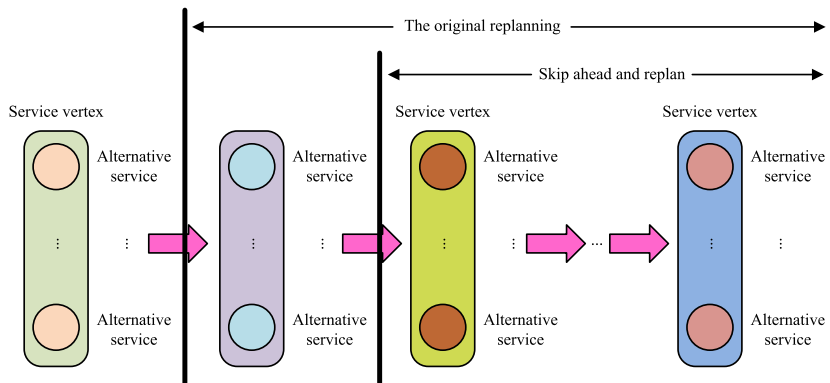


FIGURE 6. Skip forward to re-plan the rule.

In Figure 5, building a hybrid structure service process is based on a pipeline model, which is divided into five stages. Each stage contains different service composition patterns. After each stage completes running according to the pattern, it enters the next stage. Re-planning is introduced to rebind the remaining business of the process, reducing the disruption to QoS constraints. At runtime, QoS information is monitored and obtained in real-time. If a trend of briefly violating QoS constraints is detected, the re-planning is initiated for service rebinding [32]. During the re-planning process, there may be situations where each alternative service cannot function properly. Therefore, the Skip forward re-planning rule is established to skip business vertices, as shown in Figure 6.

In Figure 6, when performing the skip forward operation, the current pipeline step is directly ignored. The interval from the next step to the end of the process is entered directly. A simple strategy to handle the current failed step separately reduces the time and resource consumption of service calls. However, in environments with high uncertainty, strong QoS constraints can lead to generated service combinations that do not meet actual needs. The randomness of the environment is manifested in two aspects: the randomness of service performance and the randomness of user

selection. The service itself may face failure and recovery at any time, resulting in dynamic service availability. The hierarchical Markov decision-making for processing business processes is established, dividing the hierarchical Markov decision-making process into multiple sub processes and main processes. Markov decision process is a mathematical framework for modeling decision making in uncertain environments. In Markov decisions, the state transitions of the system depend on the current state and the actions taken, and each state transition is accompanied by a reward or cost. By calculating the expected cumulative rewards, Markov decisions can help determine the optimal strategy of action over the long term. The action set for the sub process is established, as shown in equation (10).

$$Act_{vi} = \{a_{i,j} | s_{vi,j} \in s_{vi}\} \quad (10)$$

In equation (10), Act represents the set of actions. $s_{vi,j}$ represents a single service. $a_{i,j}$ represents the request made to the corresponding service. The state transition probability is shown in equation (11).

$$\begin{cases} \Pr = 1 - q_{ud}, & \text{if } F0 = +1 \\ \Pr = q_{ud}, & \text{if } F0 = -1 \end{cases} \quad (11)$$

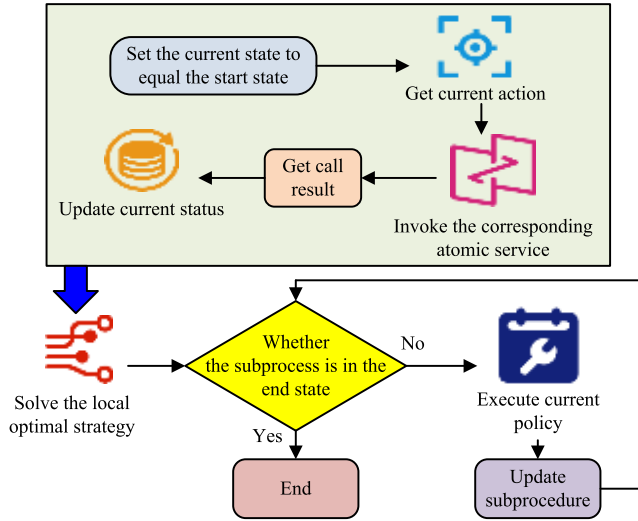


FIGURE 7. Dynamic QoS aware service composition solution process.

In equation (11), Pr represents the probability of state transition. q_{ud} represents the uncertainty of the corresponding service. F_0 represents non-zero elements in space. In the hierarchical Markov decision process, the transition probability function and state space are divided into multiple small-scale sub process Markov decision processes. The state calculation in the state space is shown in equation (12).

$$V(s) \leftarrow \text{Max}_{a \in Act} \sum_{s' \in St} \text{Pr}[\beta \text{Re}w + V(s')] \quad (12)$$

In equation (12), $V(s)$ represents the optimal value calculation result. $\text{Re}w$ represents the reward function. St represents the state space. β represents the reward coefficient. The optimal strategy generation is shown in equation (13).

$$\pi^*(s) = \arg \text{Max}_{a \in Act} \sum_{s' \in St} \text{Pr}[\beta \text{Re}w + V(s')] \quad (13)$$

In equation (13), $\pi^*(s)$ represents the optimal strategy. The business solving process is shown in Figure 7.

In Figure 7, when solving the dynamic QoS aware service composition, each sub process is solved separately, and the corresponding atomic service is called and the state is updated. The local optimal strategy of the sub process is solved, and the end state of the sub process is determined. The process is updated according to the business flow until the entire business is completed. However, the uncertainty of the service will change. The state transition function is used to update the probability of state transition, as shown in equation (14).

$$\text{Pr}(st, s, st') = \frac{\text{Pr}(st, s, \bar{st}) \cdot n_y}{n'_y}, \quad \text{where } \bar{st} \neq st' \quad (14)$$

In equation (14), n'_y is the value of adding 1 to the number of atomic services. n_y represents the number of atomic services. Two probability distributions are compared to compare the state transition functions before and after

updates. The distance between the two during the process is measured. If the distance is close, it is considered that the learning goal has been basically achieved. The study uses Kullbach-Leibler Divergence for distance calculation, as shown in equation (15).

$$KLD(p, q) = \sum_x p(x) \log \frac{p(x)}{q(x)} \quad (15)$$

In equation (15), KLD represents the Kullbach-Leibler Divergence distance. $p(x)$ and $q(x)$ represent the probability distribution of the state transition function before and after the update, respectively. Service and learning alternate to achieve long-term stable dynamic QoS aware service solving. In the actual application of the method, first, clear the service requirements in the actual scenario, and then configure the corresponding service operating environment according to the result of the requirement analysis. By using the service registry, the Web services that can meet the needs of users are discovered and registered. The QoS aware service generation method is adopted, and the weight of the performance indicators of the services is assigned based on AHP. Use a hierarchical Markov decision process to intelligently adjust service selection and composition strategies, bind business requirements to specific atomic services based on service composition policies, and execute services to complete user-specified business processes. During service execution, QoS indicators are monitored in real time. Once a trend that may violate QoS constraints is detected, the replanning mechanism is started immediately to rebind services and ensure service quality.

IV. PERFORMANCE ANALYSIS OF DYNAMIC QoS AWARE SERVICE COMPOSITION FRAMEWORK

To analyze the actual performance of the dynamic QoS aware service composition framework, experimental machines are used to test the solution quality and time, running failure rate, and multi concurrency tolerance of the method. The running performance of the method is analyzed through test results, and the actual application effect of the method is analyzed in practical application scenarios.

A. PERFORMANCE TESTING OF DYNAMIC QoS AWARE SERVICE COMPOSITION FRAMEWORK

When conducting performance tests on the research method, a desktop computer with a main frequency of 2GHz and 2Gb of running memory is used as the experimental machine, running on a Windows 7 system. The research method, Analytical Hierarchy Process-Markov Decision Processes (abbreviated as AHP-MDP) method, is compared with the integer programming method and the rounding method based on linear programming in solving quality and time analysis. The concurrent carrying capacity and failure rate are tested and compared with network analysis method and multi-attribute decision-making method. The results of quality and time analysis are shown in Figure 8.

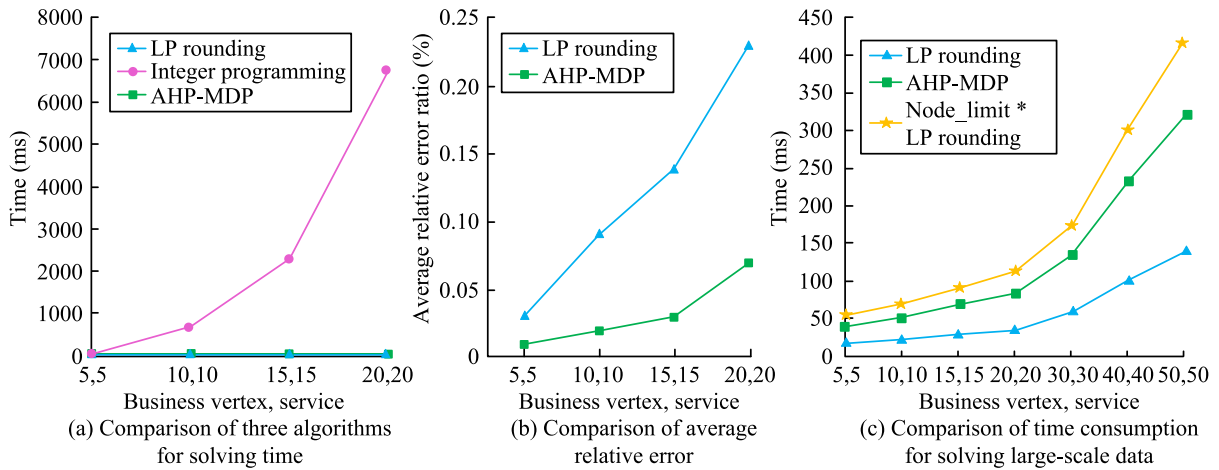


FIGURE 8. Solution quality and time analysis.

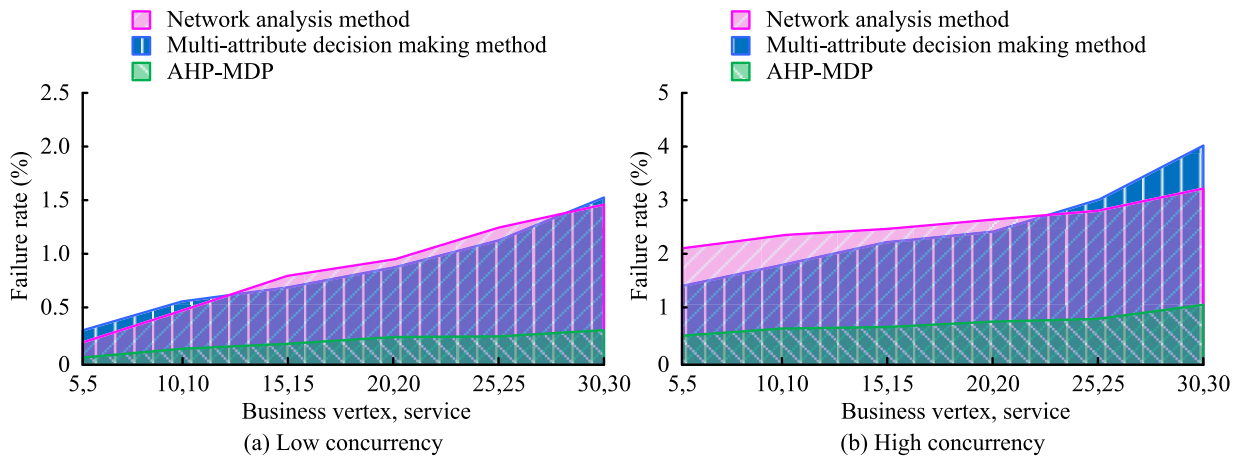


FIGURE 9. Method operation failure rate.

In Figure 8, the solution quality and time of different methods varied with the size of the problem. From Figure 8 (a), as the scale of the problem continued to expand, the time consumed by integer programming has increased rapidly, with a solving time of 6783ms at 20 business vertices. The time consumption of AHP-MDP method and rounding method based on linear programming was not significant. In Figure 8 (b), using the high-quality solution results obtained by the integer programming method as a reference, as the problem size continued to expand, the errors of the rounding method based on linear programming and the AHP-MDP method have gradually expanded. The error of the rounding method based on linear programming reached 0.227% at 20 business vertices. The error of the AHP-MDP was only 0.071%. In Figure 8 (c), in the large-scale problems, the AHP-MDP method had more calls to the upper limit of the number of nodes, resulting in a solution time between the integer programming method and the rounding method based on linear programming.

In Figure 9, the failure rates of different methods increased with the increase of problem size in both low and high concurrency scenarios. In Figure 9 (a), under low concurrency

conditions, the failure rate of the network analysis method at 5 business vertices was 0.22%. The failure rate at 30 business peaks was 1.46%. The failure rate of multi-attribute decision-making method at 5 business vertices was 0.32%. The failure rate at 30 business peaks was 1.53%. The AHP-MDP method had a failure rate of 0.08% at 5 business vertices. The failure rate at 30 business vertices was 0.31%. In Figure 9 (b), under high concurrency conditions, the failure rate of the network analysis method at 5 business vertices was 2.17%. The failure rate at 30 business peaks was 3.24%. The failure rate of multi-attribute decision-making method at 5 business vertices was 1.46%. The failure rate at 30 business peaks was 4.02%. The AHP-MDP method had a failure rate of 0.51% at 5 business vertices. The failure rate at 30 business peaks was 1.06%. This indicates that the research method can better ensure smooth operation.

In Figure 10, during the 500s multi request processing, different methods experienced crashes. Figure 10 (a) showed that the network analysis method experienced 7 crashes, with the corresponding number of concurrent users in the range of 550 to 790. In Figure 10 (b), the multi-attribute

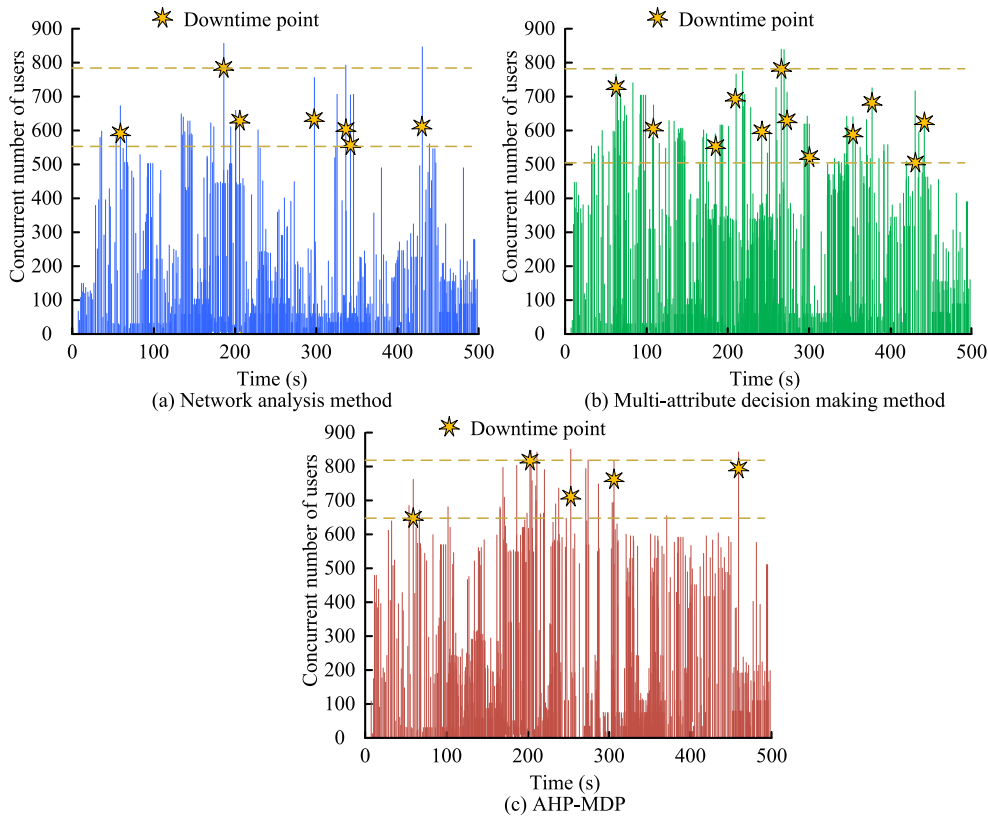


FIGURE 10. Multiple concurrent tolerance.

decision-making method experienced 12 crashes, with the corresponding concurrent users in the range of 500 to 780. In Figure 10 (c), the AHP-MDP method experienced 5 crashes, with the corresponding concurrent users in the range of 640 to 820. The AHP-MDP method has a higher lower limit of multi concurrency tolerance, which can maintain a lower incidence of downtime even after slightly exceeding the tolerance range. This indicates that the research method can simultaneously handle more requests.

B. PRACTICAL APPLICATION EFFECT ANALYSIS OF DYNAMIC QoS AWARE SERVICE COMPOSITION FRAMEWORK

When analyzing the practical application effects of the dynamic QoS aware service composition framework, the computer processor used is Intel Core i5-13490F, with a benchmark speed of 2.50GHz and 16 logic processors. The size of computer running memory is 32Gb, the memory speed is 5600MHz, Ethernet is used for Internet connection, the network type is asymmetric digital subscriber line, and the actual application operation is the data query operation in the client. The resource utilization rate during service solving is analyzed, as shown in Figure 11.

In Figure 11, the resource utilization of different devices varied during method execution. In Figure 11 (a), when using the AHP-MDP method for dynamic QoS aware service

solving, most logical processors were in a working state. Under high load conditions, all logical processors achieved an occupancy rate of over 60%, but a state of 100% occupancy hadn't occurred. From Figure 11 (b), the memory usage of the device was relatively stable during the solution, basically maintained in the range of 50% to 60%. In Figure 11 (c), during the operation, the received data of Ethernet was greater than the transmitted data. There was no continuous large amount of data transmission and reception during the operation process. This indicates that the research method can utilize device resources in a relatively stable manner, which is not cause long-term and sustained pressure on the device when its performance is appropriate.

In Figure 12, the proportion of satisfying QoS constraints increased with the number of selected services in different service availability scenarios. In Figure 12 (a), when the service availability was 1 and the number of selected services increased to 15, the proportion of satisfying QoS constraints reached 0.93. As the number of selected services continues to increase, due to the increase in the number of services in each branch, the required resources between branches has gradually approach. Even if there are prediction errors, the resource allocation is not significantly different. From Figure 12 (b), when the branch mode degenerated to the pipeline mode, the proportion of satisfying QoS constraints at service availability of 0.8 and 0.9 was consistent with the normal branch mode.

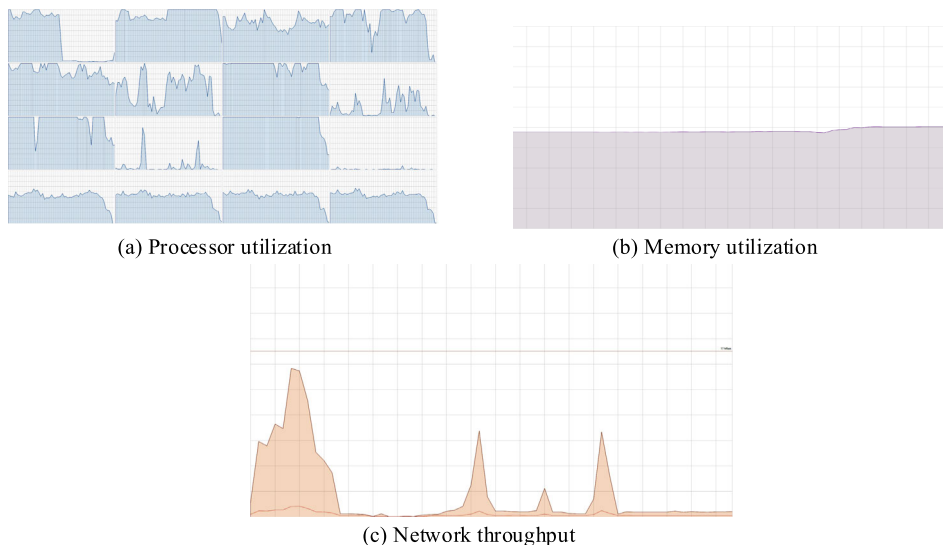


FIGURE 11. Resource utilization.

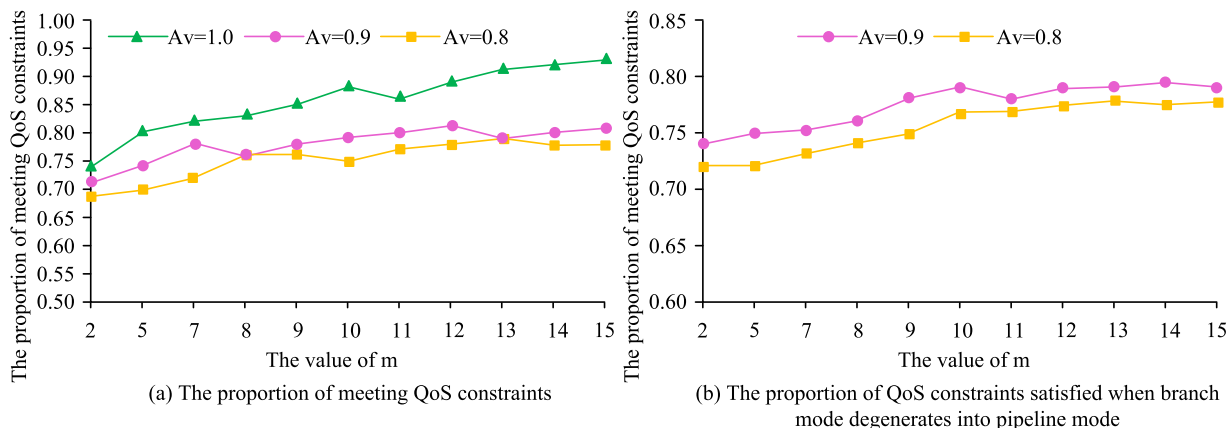


FIGURE 12. QoS constraint ratio analysis.

In Figure 13, different methods had a certain packet loss rate at runtime. In Figure 13 (a), in the absence of network fluctuations, the packet loss rate of the multi-attribute decision-making method fluctuated between 0.9% and 5.3%. The packet loss rate of network analysis method fluctuated between 6.9% and 11.4%. The packet loss rate of the AHP-MDP fluctuated between 0.1% and 0.7%. In Figure 13 (b), the packet loss rate of multi-attribute decision-making method fluctuated between 5.1% and 12.6% in the presence of network fluctuations. The packet loss rate of network analysis method fluctuated between 4.2% and 9.5%. The packet loss rate of the AHP-MDP method fluctuated between 1.6% and 2.9%. The packet loss rate of multi-attribute decision-making method and AHP-MDP method has increased, which is a normal situation. The decreased packet loss rate in network analysis method is due to the decrease in data acquisition caused by network fluctuations. This indicates that the research method has higher data transmission reliability at runtime.

In Figure 14, the power consumption performance of different methods varied at runtime, and even the power consumption performance of the same method varied in different environments. From Figure 14 (a), in an environment without network fluctuations, the operating power of the network analysis method ranged from 470W to 640W. The operating power of multi-attribute decision-making method ranged from 460W to 690W. The operating power of the AHP-MDP method was in the range of 420W to 510W. In Figure 14 (b), in an environment with network fluctuations, the operating power of the network analysis method was in the range of 520W to 720W. The operating power of multi-attribute decision-making method was in the range of 590W to 730W. The operating power of the AHP-MDP method was in the range of 430W to 530W. Different methods require more energy to combat network fluctuations, and the AHP-MDP method has a smaller increase in operating power, indicating that the research method has better operational power economy.

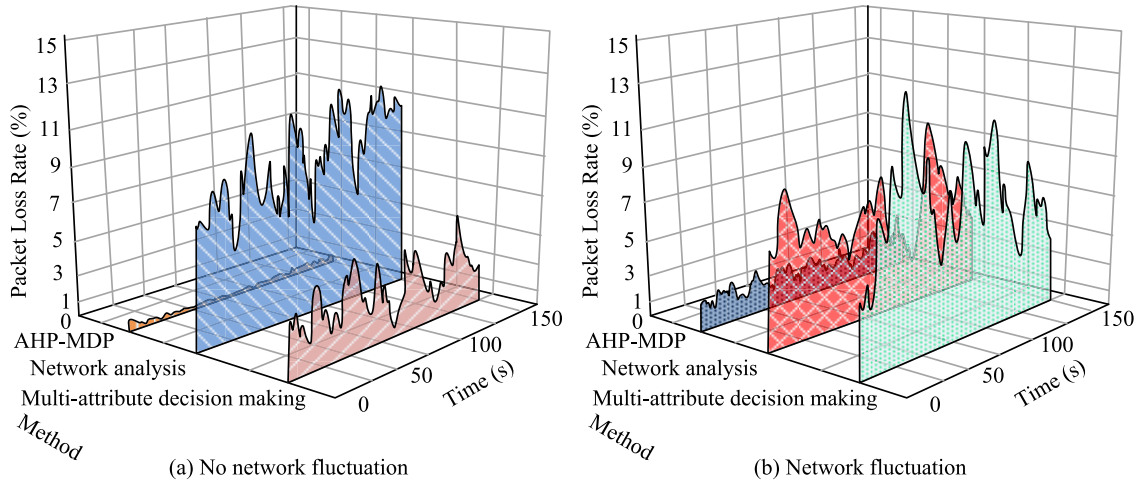


FIGURE 13. Analysis of packet loss rate during operation.

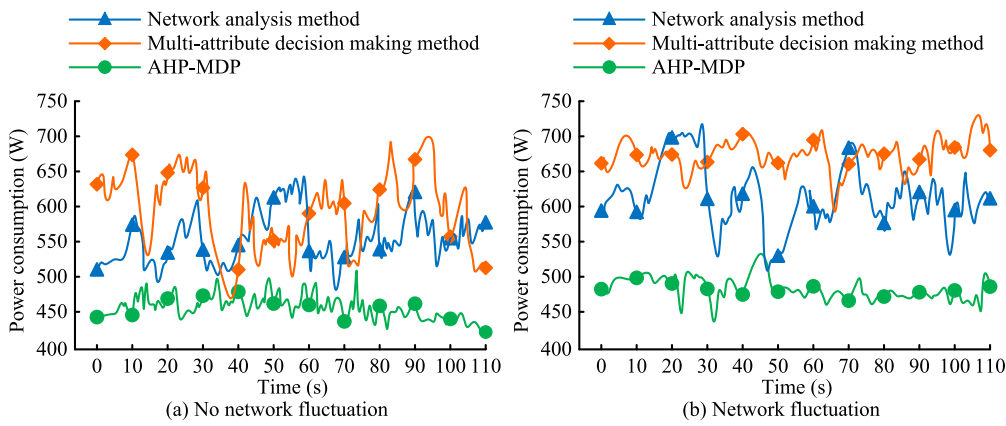


FIGURE 14. Operation power analysis.

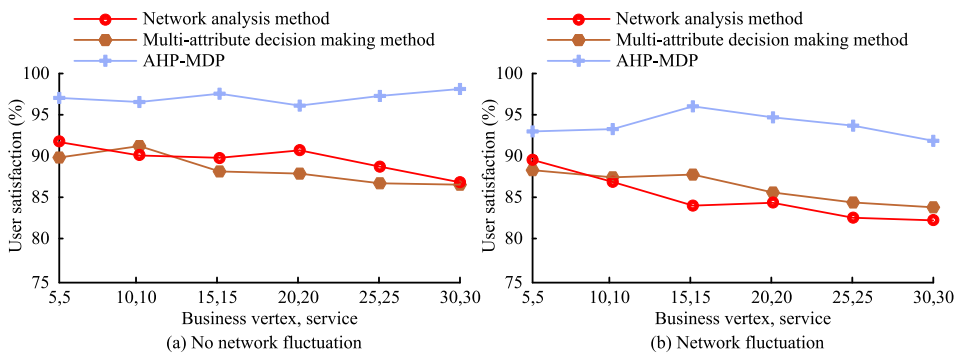


FIGURE 15. User satisfaction analysis.

In Figure 15, users had various satisfactions with solving problems of different scales generated by different methods. In Figure 15 (a), in an environment without network fluctuations, the user satisfaction of the network analysis method was 91.9% when there were 5 business vertices. When the business vertices were 30, the user satisfaction rate was 87.1%. The user satisfaction rate of multi-attribute decision-making method was 89.7% when there were

5 business vertices. When the business vertices were 30, the user satisfaction rate was 86.8%. The user satisfaction of the AHP-MDP method was 97.1% when there were 5 business vertices, and 97.9% when there were 30 business vertices. In Figure 15 (b), in an environment with network fluctuations, the user satisfaction rate of the network analysis method was 89.6% when the number of business vertices was 5. The user satisfaction rate was 82.3% when the business vertices

were 30. The user satisfaction rate of multi-attribute decision-making method was 88.1% when there were 5 business vertices, and 88.9% when there were 30 business vertices. The user satisfaction of the AHP-MDP method was 92.9% when there were 5 business vertices. When the business vertices were 30, the user satisfaction rate was 92.1%. This indicates that the research method has better practical performance, which can provide users with a more comfortable operating experience during runtime.

V. CONCLUSION

A dynamic QoS aware service composition framework based on AHP and hierarchical Markov decision was proposed to improve the management quality of Web service QoS. User preferences were processed through a user QoS preference model, taking into account the impact of each QoS. The consistency of the matrix was judged through consistency indicators, and the weights were calculated to obtain the final matching service as the executed service. The vertex set was used to represent the specific business. Then a heuristic algorithm based on Local Branching was established to quickly solve the approximate optimal solution of the problem. Hierarchical Markov decision-making was established to process business processes, and the effectiveness of the method was analyzed. According to the experimental results, in the analysis of solving time, the solving time of the research method at 20 business vertices did not exceed 500ms. In the run failure rate test, the failure rate of the research method under high concurrency conditions with 30 business vertices was 1.06%. When conducting resource utilization analysis, under high load conditions, all logical processors were not in a state of 100% occupancy. In analyzing the proportion of satisfying QoS constraints, when the service availability was 1 and the number of selected services increased to 15, the proportion of satisfying QoS constraints reached 0.93. This indicates that the research method has better performance in combining service solutions, which can make service calls in a more concise manner. Although the heuristic algorithm is used to optimize the solution process when dealing with large-scale service composition problems, the computational complexity of the algorithm may increase with the increase of the number of services, which affects the real-time performance of the solution. Future research could explore the integration of semantic technologies into service composition frameworks to enhance the semantic accuracy of service discovery and matching, and improve the quality and efficiency of service composition. At the same time, for large-scale service composition problems, more efficient algorithms or optimization strategies can be studied to reduce computational complexity and improve the scalability of solutions.

REFERENCES

- [1] M. Lauri, D. Hsu, and J. Pajarinen, "Partially observable Markov decision processes in robotics: A survey," *IEEE Trans. Robot.*, vol. 39, no. 1, pp. 21–40, Feb. 2023, doi: [10.1109/TRO.2022.3200138](https://doi.org/10.1109/TRO.2022.3200138).
- [2] V. Goyal and J. Grand-Clément, "Robust Markov decision processes: Beyond rectangularity," *Math. Operations Res.*, vol. 48, no. 1, pp. 203–226, Feb. 2023, doi: [10.1287/moor.2022.1259](https://doi.org/10.1287/moor.2022.1259).
- [3] L. Chen and Y. Deng, "An improved evidential Markov decision making model," *Appl. Intell.*, vol. 52, no. 7, pp. 8008–8017, Oct. 2022, doi: [10.1007/s10489-021-02850-0](https://doi.org/10.1007/s10489-021-02850-0).
- [4] W. Yang, L. Zhang, and Z. Zhang, "Toward theoretical understandings of robust Markov decision processes: Sample complexity and asymptotics," *Ann. Statist.*, vol. 50, no. 6, pp. 3223–3248, Dec. 2022, doi: [10.1214/22-aos2225](https://doi.org/10.1214/22-aos2225).
- [5] M. Naghdehforousha, M. D. T. Fooladi, M. H. Rezvani, and M. M. G. Sadeghi, "BLMDP: A new bi-level Markov decision process approach to joint bidding and task-scheduling in cloud spot market," *Turkish J. Electr. Eng. Comput. Sci.*, vol. 30, no. 4, pp. 1419–1438, May 2022, doi: [10.55730/1300-0632.3857](https://doi.org/10.55730/1300-0632.3857).
- [6] B. Alaya and L. Sellami, "Multilayer video encoding for QoS managing of video streaming in VANET environment," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 18, no. 3, pp. 1–19, Mar. 2022, doi: [10.1145/3491433](https://doi.org/10.1145/3491433).
- [7] M. Norouzi Shad, M. Maadani, and M. Nesari Moghadam, "GAPSO-SVM: An IDSS-based energy-aware clustering routing algorithm for IoT perception layer," *Wireless Pers. Commun.*, vol. 126, no. 3, pp. 2249–2268, Oct. 2022, doi: [10.1007/s11277-021-09051-5](https://doi.org/10.1007/s11277-021-09051-5).
- [8] M. Beshley, N. Kryvinska, H. Beshley, O. Panchenko, and M. Medvetskiy, "Traffic engineering and QoS/QoE supporting techniques for emerging service-oriented software-defined network," *J. Commun. Netw.*, vol. 26, no. 1, pp. 99–114, Feb. 2024, doi: [10.23919/jcn.2023.000065](https://doi.org/10.23919/jcn.2023.000065).
- [9] L. A. L. F. D. Costa, R. Kunst, and E. P. de Freitas, "Intelligent resource sharing to enable quality of service for network clients: The trade-off between accuracy and complexity," *Computing*, vol. 104, no. 5, pp. 1219–1231, Jan. 2022, doi: [10.1007/s00607-021-01042-5](https://doi.org/10.1007/s00607-021-01042-5).
- [10] C. Li, Y. Liu, J. Xiao, and J. Zhou, "MCEAACO-QSRP: A novel QoS-secure routing protocol for industrial Internet of Things," *IEEE Internet Things J.*, vol. 9, no. 19, pp. 18760–18777, Oct. 2022, doi: [10.1109/JIOT.2022.3162106](https://doi.org/10.1109/JIOT.2022.3162106).
- [11] Ç. K. Kaymaz, S. Birinci, and Y. Kızılcak, "Sustainable development goals assessment of erzurum province with SWOT-AHP analysis," *Environ., Develop. Sustainability*, vol. 24, no. 3, pp. 2986–3012, Mar. 2022, doi: [10.1007/s10668-021-01584-w](https://doi.org/10.1007/s10668-021-01584-w).
- [12] C. P. C. Senan, R. S. Ajin, J. H. Danumah, R. Costache, A. Arabameri, A. Rajaneesh, K. S. Sajinkumar, and S. L. Kuriakose, "Flood vulnerability of a few areas in the foothills of the western ghats: A comparison of AHP and F-AHP models," *Stochastic Environ. Res. Risk Assessment*, vol. 37, no. 2, pp. 527–556, Feb. 2023, doi: [10.1007/s00477-022-02267-2](https://doi.org/10.1007/s00477-022-02267-2).
- [13] R. T. Vilasan and V. S. Kapse, "Evaluation of the prediction capability of AHP and F-AHP methods in flood susceptibility mapping of ernakulam district (India)," *Natural Hazards*, vol. 112, no. 2, pp. 1767–1793, Mar. 2022, doi: [10.1007/s11069-022-05248-4](https://doi.org/10.1007/s11069-022-05248-4).
- [14] S. Duleba, "Introduction and comparative analysis of the multi-level parsimonious AHP methodology in a public transport development decision problem," *J. Oper. Res. Soc.*, vol. 73, no. 2, pp. 230–243, Feb. 2022, doi: [10.1080/01605682.2020.1824553](https://doi.org/10.1080/01605682.2020.1824553).
- [15] R. Mitra, P. Saha, and J. Das, "Assessment of the performance of GIS-based analytical hierarchical process (AHP) approach for flood modelling in Uttar Dinajpur district of west Bengal, India," *Geomatics, Natural Hazards Risk*, vol. 13, no. 1, pp. 2183–2226, Aug. 2022, doi: [10.1080/19475705.2022.2112094](https://doi.org/10.1080/19475705.2022.2112094).
- [16] R. Wang and J. Lu, "QoS-aware service discovery and selection management for cloud-edge computing using a hybrid meta-heuristic algorithm in IoT," *Wireless Pers. Commun.*, vol. 126, no. 3, pp. 2269–2282, Oct. 2022, doi: [10.1007/s11277-021-09052-4](https://doi.org/10.1007/s11277-021-09052-4).
- [17] S. Jothi and A. Chandrasekar, "An efficient modified dragonfly optimization based MIMO-OFDM for enhancing QoS in wireless multimedia communication," *Wireless Pers. Commun.*, vol. 122, no. 2, pp. 1043–1065, Jan. 2022, doi: [10.1007/s11277-021-08938-7](https://doi.org/10.1007/s11277-021-08938-7).
- [18] V. Hayyolalam, S. Otoum, and Ö. Özkasap, "Dynamic QoS/QoE-aware reliable service composition framework for edge intelligence," *Cluster Comput.*, vol. 25, no. 3, pp. 1695–1713, Mar. 2022, doi: [10.1007/s10586-022-03572-9](https://doi.org/10.1007/s10586-022-03572-9).
- [19] Y. Jiang, F. Yang, Z. Tang, and Q. Li, "Admission control of hospitalization with patient gender by using Markov decision process," *Int. Trans. Oper. Res.*, vol. 30, no. 1, pp. 70–98, Jan. 2023, doi: [10.1111/itor.12931](https://doi.org/10.1111/itor.12931).

- [20] N. Bäuerle and A. Glauner, "Distributionally robust Markov decision processes and their connection to risk measures," *Math. Operations Res.*, vol. 47, no. 3, pp. 1757–1780, Aug. 2022, doi: [10.1287/moor.2021.1187](https://doi.org/10.1287/moor.2021.1187).
- [21] K. Bhosle and V. Musande, "Evaluation of deep learning CNN model for recognition of devanagari digit," *Artif. Intell. Appl.*, vol. 1, no. 2, pp. 114–118, Feb. 2023, doi: [10.47852/bonviewaia3202441](https://doi.org/10.47852/bonviewaia3202441).
- [22] M. Kumar, J. K. Samriya, K. Dubey, and S. S. Gill, "QoS-aware resource scheduling using whale optimization algorithm for microservice applications," *Software: Pract. Exper.*, vol. 54, no. 4, pp. 546–565, Apr. 2024, doi: [10.1002/spe.3211](https://doi.org/10.1002/spe.3211).
- [23] A. Rodriguez-Valencia, J. A. Vallejo-Borda, G. A. Barrero, and H. A. Ortiz-Ramirez, "Towards an enriched framework of service evaluation for pedestrian and bicyclist infrastructure: Acknowledging the power of users' perceptions," *Transportation*, vol. 49, no. 3, pp. 791–814, Jun. 2022, doi: [10.1007/s11116-021-10194-4](https://doi.org/10.1007/s11116-021-10194-4).
- [24] G. A. Barrero and A. Rodriguez-Valencia, "Asking the user: A perceptual approach for bicycle infrastructure design," *Int. J. Sustain. Transp.*, vol. 16, no. 3, pp. 246–257, Mar. 2022, doi: [10.1080/15568318.2020.1871127](https://doi.org/10.1080/15568318.2020.1871127).
- [25] W. Zheng, M. Yang, C. Zhang, Y. Zheng, Y. Wu, Y. Zhang, and J. Li, "Application-aware QoS routing in SDNs using machine learning techniques," *Peer-to-Peer Netw. Appl.*, vol. 15, no. 1, pp. 529–548, Jan. 2022, doi: [10.1007/s12083-021-01262-8](https://doi.org/10.1007/s12083-021-01262-8).
- [26] I. Azam, M. B. Shahab, and S. Y. Shin, "Energy-efficient pairing and power allocation for NOMA UAV network under QoS constraints," *IEEE Internet Things J.*, vol. 9, no. 24, pp. 25011–25026, Dec. 2022, doi: [10.1109/JIOT.2022.3195197](https://doi.org/10.1109/JIOT.2022.3195197).
- [27] M. Motte and H. Pham, "Mean-field Markov decision processes with common noise and open-loop controls," *Ann. Appl. Probab.*, vol. 32, no. 2, pp. 1421–1458, Apr. 2022, doi: [10.1214/21-aap1713](https://doi.org/10.1214/21-aap1713).
- [28] F. Li, F. Jörg, X. Li, and T. Feenstra, "A promising approach to optimizing sequential treatment decisions for depression: Markov decision process," *Pharmacoeconomics*, vol. 40, no. 11, pp. 1015–1032, Sep. 2022, doi: [10.1007/s40273-022-01185-z](https://doi.org/10.1007/s40273-022-01185-z).
- [29] C. Mohanadevi and S. Selvakumar, "A QoS-aware, hybrid particle swarm optimization-cuckoo search clustering based multipath routing in wireless sensor networks," *Wireless Pers. Commun.*, vol. 127, no. 3, pp. 1985–2001, Dec. 2022, doi: [10.1007/s11277-021-08745-0](https://doi.org/10.1007/s11277-021-08745-0).
- [30] F. Kaviani and M. Soltanaghaei, "CQARPL: Congestion and QoS-aware RPL for IoT applications under heavy traffic," *J. Supercomput.*, vol. 78, no. 14, pp. 16136–16166, Apr. 2022, doi: [10.1007/s11227-022-04488-2](https://doi.org/10.1007/s11227-022-04488-2).
- [31] Y. Cai, X. Zhang, S. Hu, and X. Wei, "Dynamic QoS mapping and adaptive semi-persistent scheduling in 5G-TSN integrated networks," *China Commun.*, vol. 20, no. 4, pp. 340–355, Apr. 2023, doi: [10.23919/JCC.fa.2022-0548.202304](https://doi.org/10.23919/JCC.fa.2022-0548.202304).
- [32] P. Sohal, R. Tabish, U. Drepper, and R. Mancuso, "Profile-driven memory bandwidth management for accelerators and CPUs in QoS-enabled platforms," *Real-Time Syst.*, vol. 58, no. 3, pp. 235–274, Apr. 2022, doi: [10.1007/s11241-022-09382-x](https://doi.org/10.1007/s11241-022-09382-x).



RUI WANG was born in Jining, Shandong, in July 1979. She received the bachelor's degree in computer science and technology from Northwest Minzu University, in 2001, and the combined master's and Ph.D. degree in computer software and theory from Shandong University, in 2010.

From July 2010 to June 2013, she was a Senior Business Manager of informatization at China National Building Materials Group Company Ltd. From July 2013 to December 2019, she was the

Minister of the Department of Information, Beijing Mingtianhang Technology Company Ltd. Since 2020, she has been a full-time Teacher with North China Institute of Aerospace Engineering. She has published ten academic papers and six research projects. Her research interests include service computing and collaborative computing.

• • •