QoS Correlation-based Service Composition Algorithm for Multi-constraint Optimal Path

Jian Yu 1, 2 Zhixing Lin 1, 2 Qiong Yu 3 Xiangmei Xiao 1, 2
1Network Center of Sanming University, Sanming Fujian, 365004, China.
2School of Information Engineering, Sanming University, Sanming 365004, China.
3Sanming No.2 High School, Sanming Fujian, 365000, China
* Corresponding Author: Jian Yu. Email: 20020127@fjsmu.edu.cn
Received:

Abstract: With the development of network service integration, in order to obtain a better quality of service (QoS) guarantee, aiming at the characteristics of integrated network service composition and correlation, this paper proposes a QoS correlation-based service composition (QCSC) approximate algorithm based on multi-constraint optimal path selection (MCOPS). It analyses the QoS correlation criteria, correlation ratios, and Skyline algorithms to calculate the optimal path by dynamic programming, record the path nodes, and obtain the optimal service composition path that meets the user’s demand. Simulation results demonstrate the good performance of the proposed algorithm in both the average calculation time and the solution path quality. Accordingly, it can meet the QCSC requirements in the cloud environment.

Keywords: Multi-constraint, Optimal path, QoS, Correlation, Service composition

I. INTRODUCTION

With the rapid development of cloud services [1], heterogeneous networks [2], 5G networks [3], mobile payments [4], and other network services, a certain correlation can be observed between various services [5]. Cloud computing [6] employs the Internet as a carrier to provide users fast, reliable, and quality of service (QoS) guaranteed data processing services. In this complex network environment, everything is for service. Users with a mobile phone connected to the Internet can choose different services and forms anytime and anywhere. Different network services have a certain correlation in real-life scenarios, making a service correlation performance between network QoS assurance to obtain a superior service composition solution. How to deal with this correlation has become a hot research topic.

In real life, people maximize their interests based on the relevance of things. For example, when a user wants to travel, he should formulate the most suitable itinerary by correlating between various services in a certain APP on the mobile phone. For each service class (also called task or abstract service, such as ticket reservation, car-hailing service, hotel accommodation, and catering reservation), there are many candidate services (also called service instances or specific services) with different qualities to perform the functions of each service class. Therefore, people usually choose the best service composition plan based on the candidate service quality to improve the quality of experience.

Different service providers cooperate and launch a series of preferential services to attract more customers and improve service quality. For example, an APP service provider establishes a cooperative relationship with an offline entity, and passengers can enjoy the accommodation discounts while staying in their partner hotel. At this time, customers can select relevant candidate services according to their needs to obtain a superior service composition plan and improve the quality of experience. When a user selects a certain network service, its service quality often changes depending on the network service quality; that is, the choice of one candidate service can affect the service quality of the other one. This means a correlation, also known as dependency [7], between different services.

II. RELATED WORK

Based on [8], this paper employs auxiliary graphs to establish the additional search space (extra space brought by correlation) to study the correlation. Therefore, the key point is to construct an effective auxiliary graph to reflect the correlation between services.

This paper employs the method proposed by Yu et al. [16] to transform the travel plan flow chart into a candidate service network topology diagram according to the following rules to model the additional search space caused by the QoS correlation:

1. In the abstract flow chart, if there is a directed edge from the abstract service \( S_A \) (service class) to another abstract service \( S_B \), then each specific service (candidate service or service instance) in the abstract service \( S_A \) has an edge pointing to each specific service in the abstract service \( S_B \).

2. The cost of each specific service is set on the corresponding service node.

3. A source node \( s \) and a target node \( t \) are added, where the node \( s \) connects all specific service nodes in the first abstract service, while the network delay value of these edges is set to 0, and the node \( t \) connects all specific service nodes in the last abstract service.

4. The network delay value of each specific service is moved to its outgoing edge.

After the above four composition steps, a new directed acyclic service network graph can be obtained, in which each node has a network service cost value and each edge has a network delay value. As shown in Fig. 1, the QCSC problem can be modeled as an MCOPS problem. Finding a path from \( s \) to \( t \) in the directed graph that meets user requirements is a feasible service composition plan.
The above application examples only consider cloud costs while selecting the optimal SCP. When there is a correlation between the QoS of network services (such as the network delay), the service composition becomes more complicated. According to Table 1, the SCP obtained after considering the QoS correlation between network services can be seen from the data in this table that as the service cost decreases, the network delay increases. Thus, considering the QoS correlation between cloud and network services is a multi-constraint service composition problem. Therefore, it is vital to provide an algorithm that can deal with the multi-constraint service composition problem to effectively improve users' QoS.

| Service path | Service cost (yuan) | Network delay (ms) |
|--------------|--------------------|--------------------|
| p₁           | 540                | 3.3                |
| p₂           | 460                | 2.9                |
| p₃           | 450                | 2.8                |

### III. QCSC ALGORITHM

**Composition service based on Skyline algorithm**

Skyline algorithm\[^9\] aims to filter out the representative data set, which is not dominated by other data from a large data set. This algorithm is applied to reduce the number of candidate service sets and reduce the time complexity of the service composition algorithm. When the other composition services do not dominate a single one, it belongs to Skyline. Since the Skyline composition service contains all possible "optimal" services, using Skyline provides the composition service agent to respond to user requests quickly. Each Skyline composition service corresponds to a certain combination of user preferences (the set of weights that users assign to each QoS attribute). If the user determines this combination of preferences, the composition service will be returned to the user. For non-Skyline composition services, no matter what weight the user determines, they will not be selected. Therefore, the composition service agents do not have to perform global search calculations. However, they should search for a composition service that satisfies the user's constraints and has the optimal QoS in the Skyline composition service, thereby improving efficiency.

**A. QCSC algorithm based on the MCOP**

The QCSC aims to eliminate the QoS correlation between network services and users. Considering the correlation between QoS leads to a larger search space and indicates better quality service composition solutions. Constructing an auxiliary graph to express the QoS correlation can effectively simplify the problem into a multi-constraint QoS optimal path selection problem. Therefore, this paper proposes the MCOPS algorithm to obtain a network service composition plan that meets the users' QoS requirements.

The pseudo-code of the MCOPS algorithm of the network-aware service composition is shown in Algorithm 1.

---

**Algorithm 1**

**Input:** Graph \( G(V, E, s, t, δ, W, c, H, K) \), Parameter \( ϵ \), Correlative criterion

**Output:** Path \( p^\epsilon \)

1. First, construct an auxiliary graph \( G'_i(V', E') \) similar to the graph \( G(V, E) \) for each correlation criterion \( z_i(α, β) \in Z \). Then, re-weight all the outgoing edges of the correlation node in \( G' \) according to the correlation criterion. For example, if three nodes are involved in a correlation criterion in the auxiliary graph \( G' \), three new weights will be given to their outgoing edges (the outgoing edges of the same node have the same weight). In other words, a new weight \( w'_i(ν) \) will be given to a correlation node \( ν' \in a \) in a correlation criterion of the graph \( G' \).
2. Delete the outgoing edges of other nodes in the same task (service class) as the correlation node \( G'_i(V', E') \); that is, delete all outgoing edges of the node \( ν \in [S_n \setminus ν'] \).
3. If \( c_ν < c_h \), delete the incoming edges of other nodes in the same task (service class) as the correlation node \( ν' \); that is, delete all incoming edges of the node \( ν \in [S_n \setminus ν'] \).
Algorithm 1

4: Compute $\delta^k_v(e) = \left\lfloor \frac{\delta^k_e + H - 1 - 1}{W_k} \right\rfloor$, and set $\Lambda = \left\lfloor \frac{H + 1}{\epsilon} \right\rfloor$, $2 \leq k \leq K$

5: for $i = 0$ to $M$
6: for $\lambda_k = 0$ to $\Lambda$, $2 \leq k \leq K$ do
7: \[ g_i^k(\lambda_2, \ldots, \lambda_K) = 0, \quad g_i^k(\lambda_2, \ldots, \lambda_K) = \infty, \quad \forall \nu \in V \]
8: end for
9: for $\lambda_k = 1$ to $\Lambda$ do
10: for $\forall (u, v) \in E$, $\lambda_k \geq \delta^k_i(u, v)$, $\nu \neq s$ do
11: \[ g_i^k(\lambda_2, \ldots, \lambda_K) = \min \left\{ g_i^k(\lambda_2 - \delta^k_i(u, v), \ldots, \lambda_K - \delta^k_i(u, v) + \delta_i(u, v)) \right\} \]
12: if $g_i^k(\lambda_2, \ldots, \lambda_K) > g_i^k(\lambda_2, \ldots, \lambda_K - 1, \lambda_K)$ then
13: \[ g_i^k(\lambda_2, \ldots, \lambda_K) = g_i^k(\lambda_2, \ldots, \lambda_K - 1, \lambda_K), \quad 2 \leq j \leq K \]
14: end if
15: end for
16: end for
17: end for
18: if $g_i^k(\lambda_2, \ldots, \lambda_K) \leq W_i$ then
19: Find a $s-t$ path $p^i$, s.t. $\delta_i(p^i) \leq W_i$, $\delta_k(p^i) \leq \lambda_k$
20: end if
21: if $\delta_k(p^i) \leq W_k$ then
22: $p^i$ is a feasible solution, path set $PS \leftarrow p^i$
23: end if
24: if $PS \neq \emptyset$ then
25: Select the optimal path $p^{opt}$ based on QCPWD in $PS$
26: return $p^{opt}$
27: end if
28: return No feasible path, EXIT.

B. Algorithm analysis

Theorem 1. The time complexity of the MCOPS algorithm in the worst case [10] is $O\left(\frac{H}{\epsilon} M M + m M + n M + n + m + M \left(\frac{H + 1}{\epsilon}\right)^{K-1}\right)$.

Proof. Step 1 (lines 1-2), according to the QoS correlation, the required time to construct the auxiliary graph is $O(M(n + m))$; Step 2 (line 3), the required time to delete the topology node and its edges is $O((M + 1)(n + m))$; Step 3 (Lines 4-17), the required time to calculate the composition path is $O(m(M + 1) \left(\frac{H + 1}{\epsilon}\right)^{K-1})$; Step 4 (Lines 18-28), the required time to select the path is $O(K(M + 1))$. Therefore, the time complexity of the MCOPS algorithm in the worst case is $O\left(\frac{H + 1}{\epsilon} M M + m M + n M + n + m + M \left(\frac{H + 1}{\epsilon}\right)^{K-1}\right)$.

IV. SIMULATION TEST AND ANALYSIS

In order to verify the performance of the MCOPS algorithm, a comprehensive comparison simulation test is performed.

A. Test preparation

The simulation employed a set of directed acyclic graphs as the test data set to evaluate the algorithm's performance effectively. Assuming that each directed acyclic graph had ten tasks ($H=10$), there were multiple candidate services in each task, while the number of services ranging from 100 to 1000 and two QoS parameters ($K=2$) was set on each edge in the directed acyclic graph.
The first QoS parameter was randomly selected between [11, 15], while the second was randomly selected between [16, 20]. The correlation node's correlation value (new weight) was randomly selected between [5, 10]. The simulations were performed under the same test environment, the same data set, the same parameter settings, and the same data structure. Moreover, the same service composition plan was employed under the same service network node scale (SNNS) to evaluate the algorithm’s performance.

B. Test index

In order to evaluate the performance and effectiveness of MCOPS, two evaluation indicators were utilized to reflect the solution time and the solution quality of the algorithm: Average Computation Time (ACT) and QoS Correlation-based Path Weights Distance (QCPWD) were employed to verify the comprehensive performance of the algorithm.

ACT was mainly employed to evaluate the algorithm indicators in terms of the network delay. The shorter the waiting time for the network user composition plan, the better the algorithm performance. QCPWD mainly considered the QoS correlation. The distance between the weights of each dimension of the SCP and the user constraints obtained after running the algorithm independently for 30 times is:

$$QCPWD(p) = \sum_{i=1}^{n} \left[1 - \frac{\delta_i(p)}{W_i}\right]$$

QCPWD reflected the end-to-end QoS guarantee level of the SCP obtained by the algorithm for users. Obviously, the higher the QCPWD value, the better the service correlation composition path obtained after calculations.

C. Test results

This paper introduced the approximation parameter $\varepsilon$ of literature [12] into the MCOPS algorithm based on the previous analysis. In the test, different $\varepsilon$ values ($\varepsilon = 0.001, 0.01, 0.01$) were adjusted to analyze the impact of $\varepsilon$ on the QoS correlation. For the user end-to-end QoS request and the user QoS multiple constraints, the edge metric parameters $W$ were set to $W_1 = 90$ and $W_2 = 80$ in all tests to ensure that the algorithm could find a feasible service composition in all service networks. In this test, the average time of the SCP was obtained by the union of the results of the algorithm's independent operation of 30 times for different values of $\varepsilon$.

Fig. 2 shows that the MCOPS algorithm could obtain the SCP that meets the users' requirements as the network service node scale increased in the interval [100-1000]. When the service node increased, the ACT value also increased, demonstrating that it took more time to find the trend of the same path in a larger-scale service network. The MCOPS algorithm mainly constructed the search space based on the number of paths hops in the SCP, which could quickly lock the range of feasible SCPs. In the test, the higher the parameter $\varepsilon$, the lower ACT spent searching for the path. This was because the smaller the algorithm parameter $\varepsilon$, the larger the path search space, and the algorithm should spend more time to find SCPs that meet user needs.

Figs. 3 and 4 show the ACT values of the MCOPS algorithm under the same SNNC but the different correlation criteria. As the correlation ratio between services increased (20% to 100%), the MCOPS algorithm needed more time to calculate the feasible SCP, increasing the ACT. This was because when there was a certain proportion of correlations between candidate services in a service network of a certain scale, the time of the original path search space would increase after considering these correlations. With the increase of $\varepsilon$, the MCOPS algorithm would require less time to calculate the feasible SCP. Increasing the value of $\varepsilon$ played an essential role in fine-tuning the search space. When $\varepsilon$ was too large, the search space may be so small, which could not contain any feasible SPC. The value of $\varepsilon$ completely depends on the dynamic settings from the service provider or user to adapt to the constantly changing network service environment.

Fig. 5 shows the ACT values of the MCOPS algorithm under different SNNSs but the same correlation ratio. For a given correlation ratio (10%), with the increase of the scale service node scale, the ACT value of the MCOPS algorithm gradually increased, demonstrating that the MCOPS
algorithm required more time to obtain a feasible SCP. This was because the search space would increase under a correlation ratio in service scenarios of any scale. This also indicated that the MCOPS algorithm could stably and reliably find a SCP satisfying the end-to-end QoS multi-constraint under different service networks according to user needs.

![Fig. 5 ACT values under different SNNSs and the same number of the correlation criteria](image)

Fig. 5 ACT values under different SNNSs and the same number of the correlation criteria

Fig. 6 (a) and (b) show the QCPWD values of the MCOPS algorithm in solving the path quality under different SNNSs, the same correlation criterion, and the same correlation ratio. For a given number of correlation criteria (3) and correlation ratio (10%), for different scale service networks ranging in the interval [100,1000], when considering the number of correlation criteria between candidate services, Fig. 11 shows that the QCPWD value was above 0.5, which was much higher than the non-correlation MCOPS algorithm, resulting in a superior SCP. In other words, a superior SCP could be obtained considering the correlation between services.

![Fig. 6 QCPWD values under different SNNSs and the same number of correlation criteria](image)

(a) The number of correlation criteria is 3
(b) The correlation ratio is 10%

![Fig. 6 QCPWD values under different SNNSs and the same number of correlation criteria](image)

V. CONCLUSION

This paper fully studied the QCSC problem of multi-constraint paths. Firstly, the QCSC was modeled, and the auxiliary graph was then introduced to construct an additional search space to express the correlation between services. Accordingly, the QCSC problem was transformed into a special MCOP problem. Secondly, an MCOPS algorithm was proposed based on the Skyline algorithm of service composition, which could solve the QoS correlation between services. The simulation experiment demonstrated that the proposed algorithm performed well in solution time, solution quality, and QCPWD, indicating that the MCOPS algorithm could quickly and effectively find an SCP with superior quality. The performance of the MCOPS algorithm was superior to that of the ADAPT algorithm. Finally, the study on service composition algorithms also provided a good theoretical foundation for today’s Internet era.

REFERENCES

[1]. Y. Z. Zhou, D. Zhang. “Near-End Cloud Computing: Opportunities and Challenges in the Post-Cloud Computing Era,” Chinese Journal of Computers, vol. 42, no. 4, pp. 677-700, 2019.
[2]. K. Yang, X. L. Ding; J. P. Liu. “Research and application of smart edge computing in IoT,” Telecommunications Science, vol. 35, no. 2, pp. 176-184, 2019.
[3]. N. Anjum, Z. Yang, I. Khan, M. Kiran, F. Wu et al., “Efficient algorithms for cache-throughput analysis in cellular-d2d 5g networks,” Computers, Materials & Continua, vol. 67, no. 2, pp. 1759-1780, 2021.
[4]. R. Sankaran., S. Chakraborty, “Why customers make mobile payments? Applying a means-end chain approach,” Marketing Intelligence & Planning, vol. 39, no. 2, pp. 109-124, 2021.
[5]. K. L. Rao. “Research on Optimization of Service Composition Based on T-QoS Awareness,” Ph.D. dissertation. Nanjing University of Posts and Telecommunications, china, 2015.
[6]. C. L. Li; Z. H. Deng., “On the QoS of Cloud Computing,” Library & Information, vol. 4, no. 4, 1-5. 2012.
[7]. X. B. Xiong, R. Yang. Computing Arbitrary Granularities of Service Process with QoS Correlations [J], Journal of Chinese Computer Systems, vol. 39, no. 7, pp. 1565-1568. 2018.
[8]. J. Huang; Q. Duan, “Network-cloud integrated service quality assurance,” in proc Electronics Industry, Bei Jing,BJ,China,pp.237-241, 2018.
[9]. L. Xiong. “Research on the Selection Mechanism of Composite Services Supporting Semantic Association,” M.S. dissertation, Nanjing University of Posts and Telecommunications, China, 2013.
[10]. L. G. Dong, G. H. Liu, “Algorithm for Solving and Updating Combinatorial Skyline,” Computer Engineering, vol. 43, no. 6, pp. 195-201, 2017.