Journal of Computers Vol. 32 No. 5, 2021, pp. 101-114
doi:10.53106/199115992021103205008

Research of the Service Composition Optimization Model Based on Petri Net

Dan Liu[1], Shu-Wen Yao[1], Yan Zeng[2], Ke Zhao[1], Yan-An Guo[1], Li Li[1*]

[1] College of Computer Science and Technology, Changchun University of Science and Technology, Changchun, Jilin, 130022, China
ld@cust.edu.cn, ysw@mails.cust.edu.cn, mathcszk271828@foxmail.com, 2595903337@qq.com, ll@cust.edu.cn

[2] Computer Science College, Sichuan University, Chengdu, Sichuan, 610065, China
zengyan9808@126.com

Received 30 January 2021; Revised 20 April 2021; Accepted 23 April 2021

Abstract. Cloud service systems bring together a wide variety of flexible and scalable mass services. The service combination scheme which can meet the user's needs is provided to the user through the flexible and changing service mode of service combination technology. However, service combination failures often occur. In view of the above problems, this paper establishes the cloud service combination optimization model. Firstly, it is proposed that the Service Combination Optimization Petri Net models and analyzes the service selection and combination. In order to avoid local convergence, combining Service Combination Optimization Petri Net with improved Genetic Algorithm, a service combination model based on Local Search Operator Genetic Algorithm is proposed, and the legitimacy of service combination sequence is verified by Petri net. The results of experiment show that the Petri net service composition model can effectively verify the constraints of service composition and the logical rationality of the system, the Local Search Operator Genetic Algorithm effectively reduces the search space and improves the convergence rate. Compared with other algorithms, the execution efficiency and accuracy of Local Search Operator Genetic Algorithm are improved.

Keywords: cloud service combination optimization, Petri net, genetic algorithm, local search operator

1 Introduction

Cloud computing is a new computing model to build a variety of available resources in the cloud platform into scalable and on-demand services that are provided to the user through the Internet platform. However, with the rapid development of cloud computing and the continuously increased of computing resources, web service providers continued to update their products, leading to the resource pool size of cloud computing to continuously extend, which put forward higher requirements on the computing power of the service, and the resource's utilization also decreased. In order to improve resources utilization and meet the complex and changeable task requests proposed by users, service composition technology is proposed [1-5].

2 Related Works

Service combination through Quality of Service (QoS) property awareness is a classic NP-Hard problem, and Seghir F [6] proposed the Hybrid Genetic Algorithm (HGA) algorithm, which combined the Fruit Fly Optimization algorithm to reduce computational time and complexity while improving the accuracy

* Corresponding Author

of functional and non-functional global evaluation values; because QoS-aware services composition enables optimization in the shortest possible time, Karimi M B, Isazadeh A [7] proposed association rules and clustering services as well as genetic algorithms to achieve services choose and reduce the search space; Researcher Jatoth C [8] proposed an improved perceptual cloud service composition (OFACSC) for the service composition problem; Omid M [9] proposed a context model for Web service composition based on user requests, and implemented the context Perceived Web service composition framework, showing how the newly proposed framework reduces malfunction and composition time.

Therefore, this article establishes a service composition optimization model to improve the utilization of service resource in the cloud environment, and at the same time evaluate the service composition according to the Quality of Service (QoS). The main contributions of this article are as follows.

(1) This article proposes a service composition model based on Service Composition Petri Net (SCPN) to model the composition process, and then clearly describe the constraint relationship and execution logic between services.

(2) Based on the Petri net model this article proposes a service composition model of the Local Search Operator Genetic Algorithm (LSOGA). The traditional genetic algorithm is improved, and the local search operator is used to prevent the algorithm from falling into local convergence and leading to local optimal results.

3 The Service Combination Model Based on Petri Net

There are many applications in the cloud environment, through which the requests make by users can be completed efficiently. Service composition integrates a set of atomic services related to specific functions of each task to complete user requests. Due to the execution logic relationship between different services, in order to understand the relationship between services, this paper uses Petri net to model and dynamically analyze and verify the net system.

Took the travel reservation system as an example for analysis to better describe the service composition process. The execution process is shown in Fig. 1.

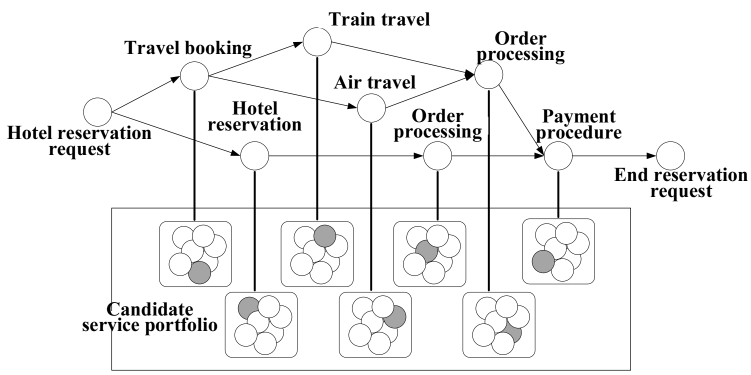

Fig. 1. Travel booking services composition diagram

The definitions of Petri net based on travel reservation service system are as follows.

Definition 3.1 defines the Travel Booking service composition process Petri net (TBPN) is composed of four tuples $TBPN = (P_{tr}, T_{tr}, F_{tr}, M_0)$. The conditions to be met are as follows:

(1) P_{tr} is a limited set of TBPN libraries, $P_{tr} = \{CS_1, CS_2, ..., CS_n\}$, SC_i represents the set of atomic service composition for subtasks.

(2) T_{tr} represents a finite set of transition in TBPN, $T_{tr} = \{\mathrm{t}_1, \mathrm{t}_2, ..., \mathrm{t}_n\}$, $\mathrm{t}_i (i = 1, 2, ..., n)$ represented the transition of control between atomic services.

(3) F_{tr} represents the set of directed arcs between libraries and transitions in TBPN, $F_{tr} = \{f_1, f_2, ..., f_n\}$.

(4) $M_0(cs_i) = 0, (i \in [1, n])$, $M_0(p_k) = 0, (k \in [1, 9])$, $M_0(p_1) = 1$.

The definition of the travel booking service combination process TBPN net and the modeling of the

Petri net TBPN, combined with each divided sub-task. All services that could complete the task were combined according to the execution logic, and each sub-task was mapped to TBPN net. The representation is shown in Fig. 2.

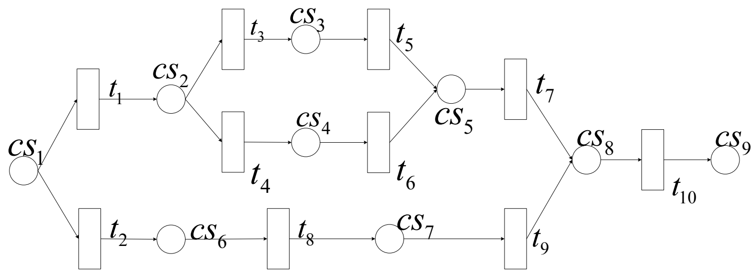

Fig. 2. Travel booking instance service combination Petri net

This article filtrate the candidate atomic services of the service composition based on objective conditions. The most basic requirement for judging whether the service composition is successful is the response time and failure frequency of the service. This paper sets up hotel reservation services to propose a Petri net-based service composition optimization model to analyze service selection and service composition. The hotel reservation service includes service library $P_2 = \{cs_6, cs_7\}$, transition set $T_1 = \{t_2, t_8, t_9\}$.

Definition 3.2 defines a Service Composition Petri Net (SCPN). It consists of the eleven tuple $P = \{P, T, E, L, I, O, F, E_{th}, L_{th}, S, M_{cs}\}$, where if and only if:

(1) $P = \{p_1, p_2, ..., p_m\}$ represents a finite set of library, where $p_i, (i = 1, 2, ..., m)$ represents the i^{th} atomic service in the set of candidate atomic services.

(2) $T = \{t_1, t_2, ..., t_n\}, (n > 0)$ represents a finite set of transition, where, $P \cap T = \varnothing$, $P \cup T \neq \varnothing$.

(3) $E = \{e_1, e_2, ..., e_m\}, (m > 0)$ represents a finite set of evaluation values for the response time of atomic services. It shows the execution efficiency of atomic services in the service combination.

(4) $L = \{l_1, l_2, ..., l_m\}, (m > 0)$ represents a finite set of atomic service failure frequencies;

(5) $I : P \times T \rightarrow \{0,1\}$ represents the output arc function of the library to the transition set.

(6) $O : T \times P \rightarrow \{0,1\}$ represents the output arc function of the transition set to the library.

(7) $F \subseteq (P \times T) \cup (P \times T)$ represents a finite set of directed arcs in the SCPN network.

(8) $E_{th} = \{e_{th\_1}, e_{th\_2}, ..., e_{th\_m}\}$ represents a finite set of critical values of service response time.

(9) $L_{th} = \{l_{th\_1}, l_{th\_2}, ..., l_{th\_m}\}$ represents a finite set of critical values of service failure frequency.

(10) $S : \{E \rightarrow E_{th}, L \rightarrow L_{th}\}$ represents the mapping relationship function from two evaluation values of the service to its critical value.

(11) $dom(F) \cup cod(F) = P \cup T$.

(12) $M : P \rightarrow \{0, 1, 2, ...\varphi\}, (\varphi \in N^+)$ represents the identification function of the Petri net SCPN. Its initial identification is M_0 represents the initial state of the Petri net. The directed arc connection between any library and the transition to prevent the appearance of outliers.

Definition 3.3 in the SCPN network, assuming that the library $p_m \in I(t_i)$ satisfies all the conditions of $M(p_i) \geq \varphi$, $e_i \geq e_{th\_i}$, $l_i \geq l_{th\_i}, (i = 1, 2, ..., m)$, then the transition t_i identifies M has the right to occur, which is expressed as $M[t_i >$.

Each library in the service composition optimization model represents a candidate atomic service. When the service response time evaluation value $e_i \geq e_{th\_i}$ and the service failure evaluation value $l_i \geq l_{th\_i}$, it means that the performance of the service has reached the basic service quality requirement for combination.

There had three subtasks of the hotel reservation process in the travel reservation service system. Suppose that the subtask $task_1$ had three candidate services p_{11}, p_{12}, p_{13}; the subtask $task_2$ had two candidate services p_{21}, p_{22}, and the subtask $task_3$ had three candidate services p_{31}, p_{32}, p_{33}. The service combination SCPN network is shown in Fig. 3.

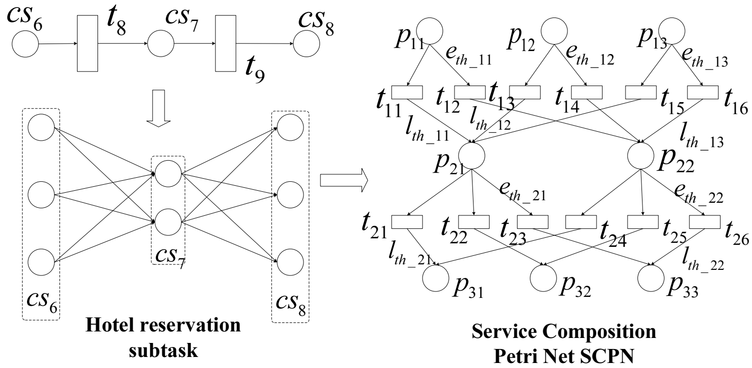

Fig. 3. Service combination Petri net SCPN

Through task division, service discovery and service selection for user requests, and analysis of the atomic service combination scheme on the SCPN net, the globally optimal service combination was selected. Calculated the QoS value of each combination scheme according to the execution logic in Fig. 1, $V_{QoS} = (P_{ava}(s), P_{mntf}(s), P_{rel}(s), T_{exe}(s), P_{suc}(s))$.

4 The Service Combination Optimization Model

This chapter uses genetic algorithm to further model the service composition based on the model built. The problem of multi-objective service composition is transformed into finding the optimal legal sequence in the Petri net SCPN model, and using genetic algorithm to continuously optimize to find the optimal solution.

4.1 Genetic Algorithm Improvement

Traditional genetic algorithms in the late evolutionary appeared convergence slows down, reducing global search capability, and in the course of evolution appear to converge to the local optimum. This paper proposes the Local Search Operator Genetic Algorithm (LSOGA) to avoid local convergence. The description of the execution flow of the local search operator is shown in Algorithm 1.

Algorithm 1. Local Search Operator (LSO).

Input: Local search step size ε_i, Search iterations T, Searched individual x_i, Control factor δ

Output: A new individual that is superior to the original

Begin

Iteration counter t= 0, the population is currently in the kth generation of evolution;

While (t $\leq$ T) do

random variable $\Delta\alpha$, $\Delta\alpha \in [-\beta_t, \beta_t]$;

$x_{next}(k)$ = $x_i(k)$ + $\Delta\alpha$;

if ($S(x_{next}(k) \bullet e_i) = True$ and $S(x_{next}(k) \bullet l_i) = True$) do

if ($F(x_i(k))$ < $F(x_{next}(k))$) do

$x_i(k)$ = $x_{next}(k)$;

end

```
        end
        else
The search results are not legitimate combination sequences or the
current solution is superior to the search results and no operation
is taken.
        end
        β_t = β_t * δ ;
        t = t + 1;
    end while
end.
```

Step1 (1-2): The relevant parameters of the algorithm execution are initialized, Local search step ε_i, search iterations *T*, searched individual x_i, Control factor δ.

Step2 (3-17): Search the local solution set for the i^{th} individual in the kth generation, control the radius of the search interval by $\Delta\alpha$ and find a new individual $x_{next}(k)$ according to the length of the radius. By judging whether the new individual $x_{next}(k)$ satisfies the specified constraints, the constraint condition for the service combination is $S(x_{next}(k)\bullet e_i) = True$, $S(x_{next}(k)\bullet l_i) = True$ defined in Petri net SCPN. The execution process (7-9) is to judge by judging the fitness function value. If the fitness value of $x_{next}(k)$ is greater than the original individual, it will be replaced, otherwise no operation will be performed. Continue to change the local search step to iterate until the end, and output new individuals.

4.2 The Service Combination Optimization Model

In the study, the improved genetic algorithm was applied to service composition and combined with Petri net SCPN to determine the legitimacy of individuals to avoid the search deviation from the optimal solution direction to solve the multi-objective service composition problem [10].

4.2.1 The Problem Code

The service composition problem is solved by binary encoding will make the problem more complicated, so decimal encoding was used to reduce the solution space. The specific coding method is as follows.

Supposed that Web request *R* requires *W* subtasks $\left(Task_1, Task_2, ..., Task_w\right)$ to be completed. Each subtask needed to be completed by a service, and each service had a candidate service set $S_i = (s_{i1}, s_{i2}, ..., s_{im}), (i = 1, 2, ..., w)$. The encoding method could be expressed as a solution space $SV = \{r_1, r_2, ..., r_k ..., r_w\}$ composed of W integers, where $r_k \in \{1, 2, ..., m\}, m \in N^*$. The number of candidate services for each service may be different, so the value of m may also be different.

4.2.2 Adaptability Function Design

Service composition is mainly to comprehensively evaluate the performance of service composition through various attribute indicators of the service. The fitness function of genetic algorithm is determined according to the transition rule of Petri net SCPN and the calculation method of each index. Because different indicators have different dimensions, each indicator needs to be dimensionless before determining the fitness function. Multiple attributes of Web service QoS have different positive and negative effects on the comprehensive value of QoS, so two service attributes are proposed: positive service attributes and negative service attributes.

(1) Positive service attributes

The availability, reliability, performance and success rate of QoS attributes are all positive service attributes. For users, web services with higher reliability have higher QoS values and are more trustworthy for users.

(2) Negative service attributes

Negative service attributes indicate that the value of the attribute is inversely proportional to the quality of service, such as execution time. The longer the execution time, the worse the user experience and the lower the service quality.

In order to unify the standard and facilitate calculation, the various attributes of QoS were standardized. Converted the value to the interval [0, 1], which the higher the QoS value, the higher the quality of service. The two-dimensional matrix representation of the standardized service quality QoS is shown in formula (1).

$$L'_{QoS} = \begin{Bmatrix} l'_{11} & l'_{12} & l'_{13} & l'_{14} & l'_{15} \\ l'_{21} & l'_{22} & l'_{23} & l'_{24} & l'_{25} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ l'_{n1} & l'_{n2} & l'_{n3} & l'_{n4} & l'_{n5} \end{Bmatrix}. \quad \textbf{(1)}$$

L'_{QoS} represents the standardized QoS matrix, and $l'_{ij},(i=1,2,..,n; j=1,2,3,4,5)$ represents the standardized service attribute index value.

$$l'_{ij} = \begin{cases} \dfrac{l_{ij} - min(l_j)}{max(l_j) - min(l_j)}, & if \quad max(l_j) \neq min(l_j) \\ 1, & if \quad max(l_j) = min(l_j) \end{cases}. \quad \textbf{(2)}$$

l_{ij} represents the unstandardized attribute index value, $min(l_j)$ represents the minimum value of the QoS attribute, l_j represents the j^{th} column attribute, and $max(l_j)$ represents the maximum value of the QoS attribute. The positive service attribute calculation method is formula (2).

$$l'_{ij} = \begin{cases} \dfrac{max(l_j) - l_{ij}}{max(l_j) - min(l_j)}, & if \quad max(l_j) \neq min(l_j) \\ 1, & if \quad max(l_j) = min(l_j) \end{cases}. \quad \textbf{(3)}$$

The calculation method of negative service attributes is formula (3).

After each quality of service attribute standardized, equation fitness function equation can be obtained as shown in (4).

$$F(x) = \begin{cases} w_1 P_{com\_av}(x) + w_2 P_{com\_mn}(x) + w_3 P_{com\_rel}(x) + w_4 P_{com\_exe}(x) + w_5 P_{com\_suc}(x), if \quad S(x.e, x.l) = true \\ 0, \qquad else \end{cases}. \quad \textbf{(4)}$$

w_1, w_2, w_3, w_4, w_5 is the balance weight of each factor satisfies $w_1 + w_2 + w_3 + w_4 + w_5 = 1$. The influence of each attribute on the fitness value can be adjusted. $S(x.e, x.l)$ is the inference rule of the individual based on the Petri net.

4.2.3 Service Composition Model Based on LSOGA Algorithm

Based on the description of LSOGA algorithm, model the service composition optimization model. According to the reasoning rules of the SCPN net, the legality of the service combination sequence was judged, which could save a lot of search time. The specific operation process is as follows:

Assumed that the population size was *N*, the number of subtasks divided was *M*, and the number of candidate services for its subtasks is *T*. Randomly generate *N* individuals of length *M*, where each individual represented a service combination scheme, and each element in the vector individual took a random value [1, T]. The maximum number of iterations of the population was *K*, the mutation probability was $\eta \in [0,2]$, and the crossover probability was $\lambda \in [0.6,1]$. The i^{th} individual of the first generation is expressed in Equation (5).

$$x_i(0) = (x_{1i}(0), x_{2i}(0) ... x_{ji}(0) \cdots x_{mi}(0)). \quad \textbf{(5)}$$

$x_i(0),(i=1,2,...,N)$ represents the i^{th} individual of the 0th generation, that is, the i^{th} service combination scheme. $x_{ji}(0),(i=1,2,...,N; j=1,2...,M)$ represents the atomic service selected by the j^{th} subtask in the i^{th} service combination scheme.

Calculated the fitness function value for each individual in the population, and determined the selection probability according to the calculated fitness value. According to the probability, the individuals who met the conditions are selected and inherited to the next generation. The probability calculation formula is shown in (6).

$$P_s(i)=\frac{F(x_i)}{\sum_{j=1}^{n}F(x_j)}. \tag{6}$$

Genetic algorithm introduces crossover operations to increase the diversity of the population. Cross-probability selected two individuals $x_i(k)$ and $x_j(k)$ performs cross-operation to produce two new service combination sequences $x_l(k+1)=(x_{1l}(k+1),\ x_{2l}(k+1),...,x_{ml}(k+1)),\ x_h(k+1)=(x_{1h}(k+1), x_{2h}(k+1),...,x_{mh}(k+1))$. Finally, the new sequences are tested for legitimacy by Petri net SCPN, which were reserved or discarded.

The combination sequence was used as an individual of the population to realize the effective combination of SCPN and genetic algorithm. The individual's gene bit was mutated. The gene bits are selected according to mutation probability η and population diversity was increased by increasing the status marker. In the process of variation operation, the probability of variation η should be reasonably set. The probability value of variation was too small, which reduced the local search capability, and the value was too large to degenerate into a random search. Finally, the new sequences are tested for legitimacy by Petri net SCPN.

The service combination model based on LSOGA algorithm performs as follows.

```
Input: The population size is N, the number of Web services is M,
       and the maximum number of iterations is K
Input: The probability of variation is η∈[0,2], cross probability
       factor is λ∈[0.6,1]
Output: The optimal scenario set for the service combination
Begin
Number of iteration k = 0, positive integer i = 1, j=1;
  for (i to N) do
    for (j to M) do
          x=x_j_min + rand(0,1)*(x_j_max - x_j_min);
    end
  end
  while (k≤K) do
       if (S(x.e, x.l)=true) do
     According to the formula (4) to calculate the individual
adaptability value;
       end
      The random number is compared with the cross probability, and
the cross operation is performed to produce two new individuals.
      The newly generated excitation sequence is validated by SCPN
       if (x_l(k+1) legitimate or x_h(k+1) legitimate) do
           New individuals are inherited to the next generation
       end
       Perform variation operations and verify legitimacy
       k = k + 1;
      Local search is performed according to algorithm 1
   end while
   return
```

```
The optimal scenario set for the service combination
end.
```

Step1 (1-7): Set population size and input parameters. Initialize the population and set the initial iteration value.

Step2 (8-11): The adaptability value is calculated for the initial population individual. The selected probability is calculated based on the adaptability value, and the individual iteration is selected by the probability size to the next generation.

Step3 (12-16): The selected individuals are cross-operated according to the cross probability, and the newly generated combination sequence is tested legitimacy according to SCPN.

Step4 (17-18): Perform population variation operations while verifying legitimacy.

Step5 (19-22): Call the local search solver to optimize the local optimal solution, determine whether the population evolution meets the conditions, and output the optimal scheme set.

5 Analysis of Experimental Results

In this paper, Petri net TBPN is dynamic performance analysis and system service execution logic deadlock judgment by using Petri's reachable marking graph analysis method [11]. In order to obtain the global optimal service combination, on the basis of Petri net SCPN describing multi-constraint combined with the LSOGA algorithm proposed in this paper to avoid the problem of high complexity or high randomness. The feasibility and validity of the model are analyzed by experimental simulation.

5.1 Experimental Data

The data set QWS DataSet is a Web service dataset collected by Professor Eyhab Al-Masri of Guelph University, which is a real data set from various service sites [12]. The dataset measures the QoS properties of the service. The availability, reliability, throughput, execution time, and success rate attributes in the data set are extracted, and some of the data is shown in Table 1. According to the QWS data set to give each sub-task different quality properties, build the corresponding candidate service set for multi-group comparison experiments.

Table 1. QoS-related properties data set

Availability (%)	Reliability (%)	Throughput (invokes/s)	Execution time (ms)	Success rate (%)
85	53	4.5	269.83	86
84	60	12.2	134.07	85
86	73	6	67.5	86
80	53	2.3	131.57	80
88	73	1.6	213.2	96
83	83	10.4	1360	84
86	73	0.7	108	95
72	73	13.3	50	72
71	83	3	1069.5	72
83	73	14.3	132	84
83	83	15.2	408	84
46	78	3.8	173	47
86	53	1.2	320.48	86
91	67	12.4	128.31	97
97	58	1.2	259	99

5.2 Comparative Experimental Analysis

In this paper, a lot of experiments are carried out around the optimization of service combination. In the course of the experiment, the Local Search Operator Genetic Algorithm (LSOGA) is compared with the Genetic Algorithm (GA) and Petri net connection method (PNFC) to verify the efficiency of the LSOGA algorithm, and the LSOGA algorithm and GA algorithm are compared with each other to verify the performance of the LSOGA algorithm. The experimental results show that the service combination model based on LSOGA algorithm is effective.

(1) The service combination of Petri net system is verified and analyzed

The reachable state of the TBPN net is analyzed by the reachable marking graph. The initial identity state of the instance is $M_0 = \{1, 0, 0, 0, 0, 0, 0, 0, 0\}$, and the reachable set as shown in Table 2. There are nine state identities, and the model is identified as $P_{tbpn} = (p_1, p_2, p_3, p_4, p_6, p_7, p_8, p_9)$, each row representing an identity.

Table 2. The reachable set of BTPN net

p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9
1	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0
0	0	1	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0
0	0	0	0	0	0	1	0	0
0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	1

The TBPN net reachable state analysis results are shown in Fig. 4.

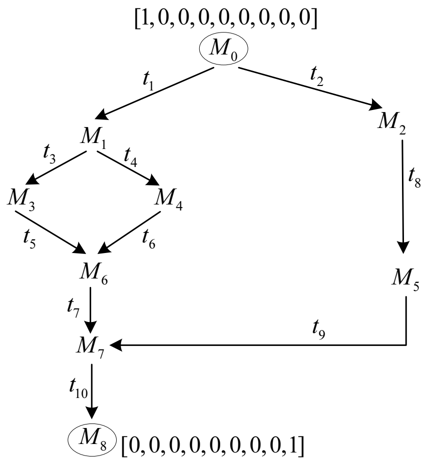

Fig. 4. The reachable state analysis result graph

The reachable status marking graph of TBPN is analyzed. The following results are analyzed using the status sequence $(M_0, M_1, M_4, M_6, M_7, M_8)$ as an example. *Stat* represents the state, *Tran* represents the transition.

(1) $M_0 = \{1, 0, 0, 0, 0, 0, 0, 0, 0\}$, $Stat = \{M_0\}$, $Tran = \varnothing$;

(2) $M_1 = \{0, 1, 0, 0, 0, 0, 1, 0, 0\}$, $Stat = \{M_0, M_1\}$, $Tran = \{t_1\}$;

(3) $M_4 = \{0, 0, 0, 1, 0, 0, 0, 0, 0\}$, $Stat = \{M_0, M_1, M_4\}$, $Trasition = \{t_1, t_4\}$;

(4) $M_6 = \{0, 0, 0, 0, 0, 0, 1, 0, 0\}$, $Status = \{M_0, M_1, M_4, M_6\}$, $Trasition = \{t_1, t_4, t_6\}$;

(5) $M_7 = \{0, 0, 0, 0, 0, 0, 0, 1, 0\}$, $Status = \{M_0, M_1, M_4, M_6, M_7\}$, $Trasition = \{t_1, t_4, t_6, t_7\}$;

(6) $M_8 = \{0, 0, 0, 0, 0, 0, 0, 0, 1\}$, $Status = \{M_0, M_1, M_4, M_6, M_7, M_8\}$, $Trasition = \{t_1, t_4, t_6, t_7, t_{10}\}$.

The result of the operation is that the system end state M_8 can always be reached from the initial state M_0 of the system through the transition sequence, hence there is no deadlock in the network system. Because all states contain all libraries and transition, the network system is active and fully covered, and the logical structure between service combinations is correct and valid.

(2) Service combination model analysis based on LSOGA algorithm

Before validating the service combination model based on LSOGA algorithm, the data set needs nondimen- sionalizetion. The experimental parameters are set to make them more convincing. Based on the data set QWS, the quality attributes of the service are standardized and normalized into intervals [0, 1]. Some of the experimental data are shown in Table 3.

Table 3. Standardize QoS quality of service attributes

Availability (%)	Reliability (%)	Throughput (invokes/s)	Execution time (ms)	Success rate (%)
0.839	0.714	0.370	0.090	0.946
0.882	0.714	0.030	0.663	0.957
0.978	0.607	0.277	0.018	1.000
0.860	0.714	0.042	0.014	0.946
0.785	0.607	0.037	0.014	0.793
0.978	0.607	0.028	0.044	0.989
0.742	0.482	0.063	0.020	0.739
0.903	0.607	0.353	0.013	0.967
0.957	0.607	0.312	0.011	0.989
0.849	0.714	0.177	0.019	0.946
0.892	0.357	0.251	0.037	0.967
0.849	0.714	0.372	0.016	0.848
0.839	0.357	0.102	0.047	0.848
0.828	0.482	0.281	0.020	0.837

The experimental configuration parameters are shown in Table 4.

Table 4. The configuration parameters of experimental

Configuration parameters	Value
Subtastay settings	10
Maximum candidate service/subtastay	100
The critical value of SCPN net E_{th}	0.8
The critical value of SCPN net L_{th}	0.3
The cross probability of the LSOGA algorithm	0.6
The variation probability of the LSOGA algorithm	0.9

① The analysis of algorithmic efficiency

The evaluation criterion is CPU running program time, with different iterations, different number of candidate services design comparison test. The increase in the number of candidate services increasing the size of the search space and leading to an increase in scale, so the efficiency of the model is validated from the runtime perspective. The average of the results of 10 executions is used as the result of the experiment to ensure credibility. The population size is 200, the number of candidate services is 100, 120, 140, 160, the number of iterations is 200, 300, 400, 500, 600. CPU execution program time experimental results are shown in Fig. 5.

The size of the population also has some influence on the efficiency of model execution. The number of candidate services is 120, and the iterations is 200, 300, 400, 500, 600 when the population size is 100, 200, 300, 400.The execution time experiment results are shown in Fig. 6.

It can be found through above two experimental results that with the number of candidate services and populations increase, the execution time also increase. The execution time growth rate is very slow and is counted in milliseconds, so the service portfolio model is efficient.

The service combination model based on LSOGA algorithm combines Petri net SCPN to greatly reduce the search space. The LSOGA algorithm is compared with the GA algorithm and PNFC to verify the efficiency of the model.

The population size of LSOGA and GA algorithms is set at 200, the number of iterations is 100, and the number of candidate services is 100, 120, 140, 160, 180 and 200.The experimental results are shown in Fig. 7.

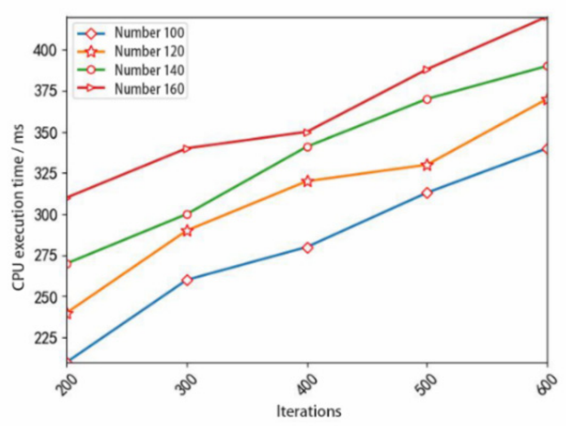

Fig. 5. Execution time for different numbers of candidate services

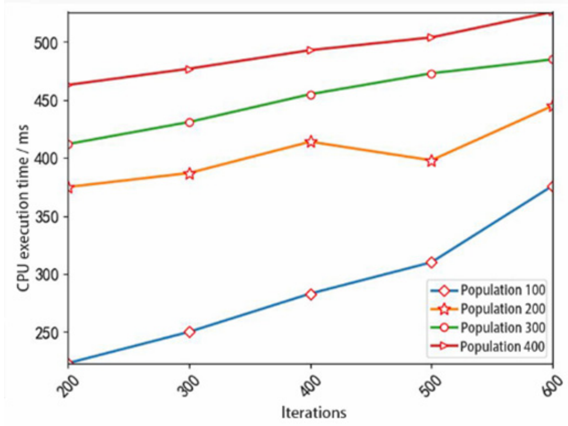

Fig. 6 Execution time for different population sizes

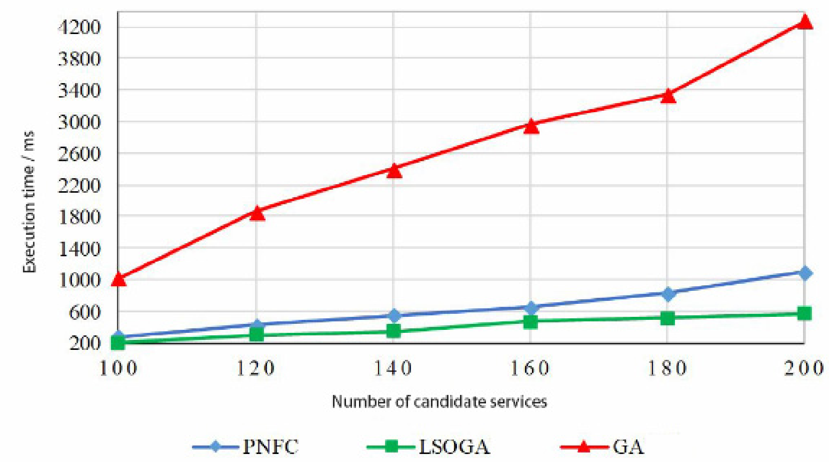

Fig. 7. Execution time for three methods of different services

After further analysis, it is found that the execution time of the algorithm proposed in this paper is shorter than that of GA and PNFC algorithm, and its growth is the slowest. It proves the efficiency of the algorithm.

② The comparison experimental analysis of algorithmic performance

Compare the feasible solution ratio of combined sequence with GA algorithm. Set the population size of 200, the number of candidate services 120, the number of iterations 200, 300, 400, 500, 600, the two methods to find a possible solution ratio of the service combination scheme. The results of the experiment are shown in Fig. 8.

Set the population size is 200, the number of iterations is 300, the number of candidate services are 100, 120, 140, 160, 180, 200. The two methods look for a feasible solution ratio for the service combination scheme. The results of the experiment are shown in Fig. 9.

Set the number of population iterations is 300, the number of candidate services is 120, the population size are 100, 200, 300, 400, 500 in the case of two methods to find a feasible solution ratio of the service combination scheme. The results of the experiment are shown in Fig. 10.

Compared with the GA algorithm, Fig. 8 to Fig. 10 are the results of feasible scale calculation under different iterations, different number of candidate services and different population sizes. The experimental results show that the comprehensive spatial search coverage of the LSOGA algorithm proposed in this paper is higher, and the optimal solution can be found more comprehensively.

The calculation of the error reflects the accuracy of the solution. Compared with the GA algorithm, set the population size is 200, the number of candidate services is 120, the number of iterations are 100, 200, 300, 400, 500, 600. The results of the experiment are shown in Fig. 11.

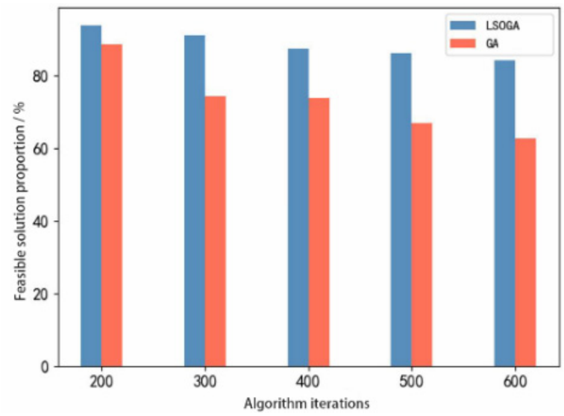

Fig. 8. Feasible solution ratio comparison under the number of different iterations

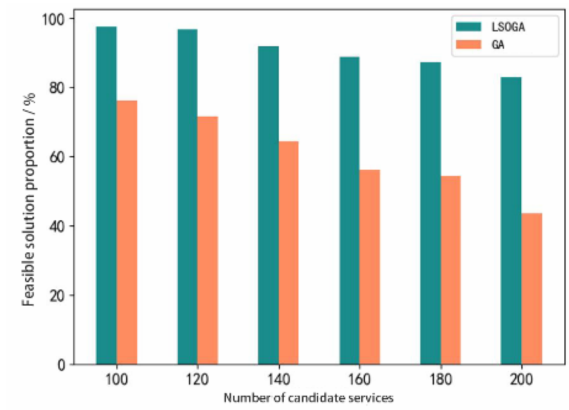

Fig. 9. Feasible solution ratios comparison under the number of different services

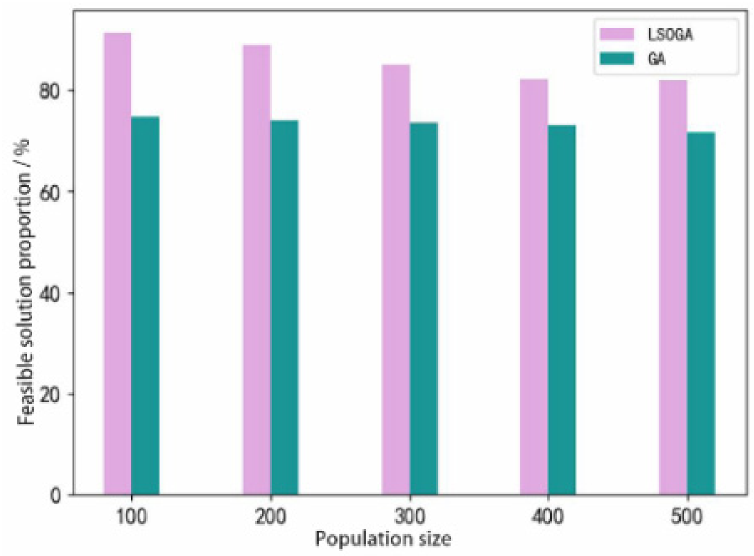

Fig. 10. Feasible solution ratios comparison under different population sizes

The number of iterations is 200, the number of candidate services is 120, and the population size are 100, 200, 300, 400, 500 to find the percentage of error of the service combination scheme with two methods. The results of the experiment are shown in Fig. 12.

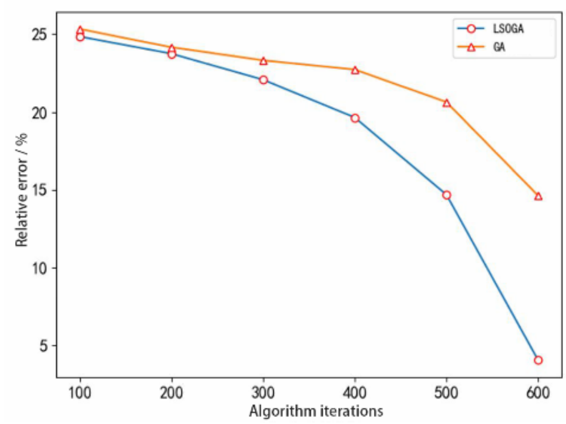

Fig. 11. Percentage comparison of errors under the number of different iterations

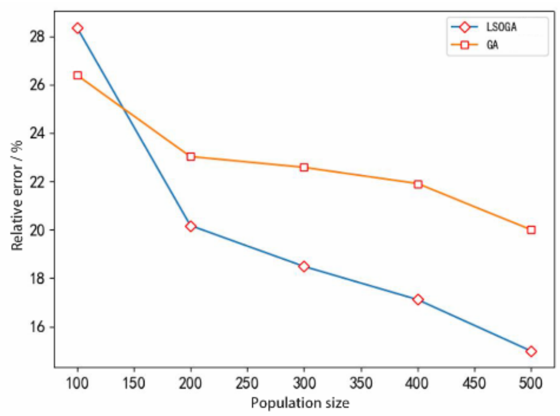

Fig. 12. Percentage comparison of errors under different population sizes

It can be seen from the experimental results of Fig. 11 that the service of GA and LSOGA algorithm has less relative error with the increase of iteration number, but the overall error of LSOGA is lower than that of GA algorithm. In Fig. 12, the error of different population sizes shows a downward trend, but the error of LSOGA decreases faster and converges faster. Therefore, the service combination model proposed in this paper solves the service combination problem more effectively.

6 Conclusion

This paper analyzes the service portfolio problem in the cloud environment and verifies the feasibility and validity of the established service portfolio model. However, there are still shortcomings. The future research direction mainly applies the algorithms of machine learning and data mining to the process of service combination.

Acknowledgements

This work was supported in part by the science and technology project of “13th Five-Year” planning of the Education Department of Jilin Province (JJKH20200801KJ); the key project of “13th Five-Year” planning of the Education Science of Jilin Province (ZD19019); Scientific research projects of higher education in Jilin Province (JGJX2020D63).

References

[1] S.-S. Peng, H.-B. Wang, Q. Yu, Multi-Clusters Adaptive Brain Storm Optimization Algorithm for QoS-Aware Service Composition, IEEE Access 8(2020) 48822-48835.

[2] Y.-F. Yang, B. Yang, S.-L. Wang, F. Liu, Y.-K. Wang, X. Shu, A dynamic ant-colony genetic algorithm for cloud service composition optimization, The International Journal of Advanced Manufacturing Technology 102(2019) 355-368.

[3] C. Jatoth, G.R. Gangadharan, R. Buyya, Optimal Fitness Aware Cloud Service Composition using an Adaptive Genotypes Evolution based Genetic Algorithm, Future Generation Computer Systems 94(2019) 185-198.

[4] A. Naseri, N.J. Navimipour, A new agent-based method for QoS-aware cloud service composition using particle swarm optimization algorithm, Journal of Ambient Intelligence and Humanized Computing 10(2019) 1851-1864.

[5] S.K. Gavvala, C. Jatoth, G.R. Gangadharan, R. Buyya, QoS-aware cloud service composition using eagle strategy, Future Generation Computer Systems 90(2019) 273-290.

[6] F. Seghir, A. Khababa, A hybrid approach using genetic and fruit fly optimization algorithms for QoS-aware cloud service composition, Journal of Intelligent Manufacturing 6(2018) 1-20.

[7] M.B. Karimi, A. Isazadeh, A.M. Rahmani, QoS-aware Service Composition in Cloud Computing Using Data Mining Techniques and Genetic Algorithm, The Journal of Supercomputing 73(4)(2017) 1387-1415.

[8] C. Jatoth, G.R. Gangadharan, U. Fiore, Optimal fitness aware cloud service composition using modified invasive weed optimization, Swarm and Evolutionary Computation 2018.

[9] M. Omid, Context-Aware Web Service Composition based on AI Planning, Applied Artificial Intelligence 31(1)(2017) 23-43.

[10] Y.-F. Yang, B. Yang, S. Wang, T. Jin, S. Li, An enhanced multi-objective grey wolf optimizer for service composition in cloud manufacturing, Applied Soft Computing 87(2020).

[11] Y.-Y. Du, Y.-H. Ning, Property analysis of logic Petri nets by marking reachability graphs, Frontiers of Computer Science 8(2014) 684-692.

[12] E. Al-Masri, Q.-H. Mahmoud, QoS-based Discovery and Ranking of Web Services, in: Proc. International Conference on Computer Communications and Networks (ICCCN), 2007.