Research on the Intelligent Composition and Selection Method for Semantic Web of Things Services

Xinying Chen*, Guanyu Li, Yueping Zhang, Yuefan Liu and Yunhao Sun

***Abstract*—In order to find an effective composition and selection algorithm for Semantic Web of Things services, concepts, "optimal semantic matching for service bipartite graph", "parameter dependency degree" and "set of Qos high-quality solutions" are defined firstly. In the meanwhile, relevant theorems are drawn out. Then combined with characteristics of the service composition and selection problem, considering influence factors: the local semantic matching between sub-services, the global semantic matching between demands and services, the dependencies between input and output parameters, and the Qos quality model for composite services, a quality evaluation model $Qos(CS)$ for composite services is proposed here. After that, a dynamic service composition and selection algorithm $IC\&S\_SWTS$ is designed based on the quality evaluation model and genetic algorithm. With consideration of above factors, the new algorithm effectively solves problems in existing algorithms and further improves precision. Finally, theoretical analysis and experimental results reveal the validity of the proposed algorithm. And the algorithm provides reasonable approximate optimal solutions at lower costs.**

***Index Terms*—semantic web of things services, semantic matching, dynamic service composition, bipartite graph matching, genetic algorithm.**

I. Introduction

IN 1999, Ashton first proposed the Internet of Things [1]. The original intention of the Internet of Things was to realize the global sharing of information about things [1]. However, due to the lack of support of relevant knowledge, the agents using the thing's information may not be able to accurately understand the received information and often be ambiguous. The introduction of semantic annotation and ontology [2] will greatly improve the ability of agents to understand and reason related information, resulting in a qualitative improvement in the function of the Internet of Things, as shown in Fig.1 of [3]. This paper refers to this smart Internet of Things as the Semantic Web of Things (SWoT). Semantic Web of Things is not a simple overlay of the Internet of Things and the Semantic Web, but an essential promotion of the Internet of Things.

Web services use standard format information such as WSDL. Using semantic web technology to extend web services, semantic web services with semantic tag information can be obtained [4]. Semantic Web of Things services and other services are not completely divided [5]. If web services are added electronic labels, Internet of Things services can be generated [6]. After adding semantic annotations to Web services and mapping them with physical entities, they have the characteristics of Semantic Web of Things services [7].

However, the distribution and heterogeneity of the Semantic Web of Things are doomed to the diversity and complexity of requirements and the massiveness and heterogeneity of services. At the same time, in Semantic Web of Things, not only services and demands are dynamic, but also environments are dynamic. Therefore, it is difficult for users to quickly obtain accurate and effective services. Semantic Web technology provides a static semantic island, and it cannot provide a global, dynamic, and effective Semantic Web services for the Internet of Things [2]. Therefore, in order to provide services that are adaptive in the Semantic Web of Things, it is particularly important to seek efficient service composition and selection methods.

Tang Yingying et al. [8] proposed a composition method of web services based on event ontology. Liu Sipei et al. [9] turned the concurrent service composition into a sequential combination based on WSC. Chang Liang et al. [10] proposed a DDL (X)-based semantic web service combination method. Liu Yisong et al. [11] proposed a new semantic web service discovery method based on clustering and bipartite graph matching technology. Kelaidonis D et al. [12] proposed a joint edge cloud IoT platform to support the combination and distribution of semantic services. Neeraj Kumar et al. [13] designed a new context-aware P2P service selection and recovery algorithm. Rodriguez-Mier P et al. [14] defined an integrated framework "ComposIT/iServe architecture" for semantic web service discovery and composition. Bansal S et al. [15] proposed a service combination technique based on generalized semantics. Puttonen J et al. [16] proposed a planning-based semantic web service composition framework. Bekkouche A et al. [17] proposed a semantic web service optimization combination method based on user-constrained QoS perception. Chattopadhyay S et al. [18] proposed a service composition framework, which can use the semantic information of web services to generate different clusters and use these clusters to build a combined solution. Paulraj D et al. [19] proposed a new atomic service discovery, composition and plan generation architecture based on the abduction event calculus. Khanouche M E et al. [20] proposed an intelligent service composition algorithm (CQCA) based on clustering and QoS awareness.

The service resources are changed in real time, and the

Manuscript received June 21, 2019; revised June 30, 2020. This work was supported by the National Natural Science Foundation of China under Grant (No.61976032), the Natural Science Foundation of LiaoNing Province (No.2019-ZD-0105 and No.20180550499) and Research Fund Project of LiaoNing Province (No.JDL2019022).

X. Y. Chen is an associate professor at School of Software Technology, Dalian Jiao Tong University, Dalian, LiaoNing, 116021 China e-mail: chenxy1979@163.com.

G. Y. Li is a professor at Faculty of Information Science and Technology, Dalian Maritime University. Y. P. Zhang and Y. F. Liu are associate professors at Dalian Jiao Tong University. Y. H. Sun is with Dalian Maritime University.

service composition and selection methods in above papers are mostly directed to static service information and cannot effectively satisfy dynamic and composite requirements. At present, with the increase of service resources, a large number of services with similar functional attributes and different Qos (non-functional) attributes have emerged. The above methods fail to comprehensively consider the functional and non-functional factors of services. Qos-based related research includes integer programming, heuristics, and genetic algorithms [21]. Qos-based dynamic service composition and selection is a multi-objective and multi-constraint NP-Complete problem [22]. The genetic algorithm is an intelligent optimization method, which has the characteristics of parallel computing and group optimization. It has been widely used to solve various NP-Complete problems and service composition and selection problems [21-31]. The research of dynamic service composition based on genetic algorithms has become an important research direction in Semantic Web of Things.

However, existing solutions [21-31] only consider the overall quality characteristics of a composite service, only based on Qos non-functional attribute values, without considering the overall validity of a composite service. When designing related algorithms, the semantic matching relationship between sub-services in a composite service and the dependency relationship between the input and output parameters of each service are not taken into consideration. Moreover, the existing solutions mostly use standard genetic algorithms, and the standard genetic algorithms use fixed mutation probability and crossover probability in an iterative process. However, the population diversity and evolution of different periods are different, and the fixed probabilities may lead to premature convergence. Therefore, the probabilities should be adjusted timely.

To sum up the above problems, taking into account the local semantic matching between sub-services in a composite service, the global matching between demands and services, the dependency relationship between the input and output parameters, and the QoS quality model for composite services, a quality evaluation model $Qos(CS)$ for composite services is designed. Then a fitness function $Fitness(CS)$ is designed based on the evaluation model. After that, based on genetic algorithm, an intelligent combination and selection algorithm $IC\&S\_SWTS$ for Semantic Web of Things services is proposed. Finally, experiments prove the validity of the algorithm.

II. Related Definitions

To facilitate the followup discussion and analysis, the following definitions are proposed based on related definitions [5].

Definition 1: (Function Operation) A function operation Fun can be formalized as a 7-tuple $Fun = (Inf$, In, Out, F_{IO}, Pre, $Post$, $Qos)$:

(1) Fun: Fun represents the label of the function operation and is unique.

(2) Inf: Inf represents the feature description information of Fun.

(3) In: In=$\{in_1, in_2, \cdots, in_m\}$ represents the input parameter set of Fun.

(4) Out: Out=$\{out_1, out_2, \cdots, out_n\}$ represents the output parameter set of Fun.

(5) Pre: Pre=$\{pre_1, pre_2, \cdots, pre_l\}$ represents a set of preconditions, including semantic contexts, environment contexts, device contexts, and user contexts and so on.

(6) $Post$: $Post$=$\{post_1, post2, \cdots, post_k\}$ represents a set of postconditions indicating the effect on the current state after the execution of Fun.

(7) F_{IO}: Represents the dependent function between output parameters and input parameters of Fun, and is a mapping from Out to the power set of In, $F_{IO} : Out \rightarrow 2^{In}$.

(8) Qos: Qos=$\{Qos_1, Qos_2, Qos_3, Qos_4\}$ represents a set of non-functional attributes of Fun, including: execution cost, execution time, availability, and reliability.

Any $pre_i \in Pre$ or $post_o \in Post$ can be modeled as a RDF triple [32]. Therefore, the semantic similarity between pre_i and $post_o$ can be obtained according to the semantic similarity between the triples [33]. The label, attributes, parameters, and conditions of Fun mentioned above are all described using domain ontology.

Definition 2: (Service) An abstract service S can be formalized as a 2-tuple $S = (INF, FUN)$.

(1) S: S represents the label of the service and is unique.

(2) INF: INF represents the feature description information of S.

(3) FUN: FUN=$\{Fun_1, Fun_2, \cdots, Fun_w\}$ represents a set of function operations provided by S.

Definition 3: (Service Request) A service request R can be formalized as R=(In^R, Out^R, Pre^R, $Post^R$, W).

W is a threshold vector, $\forall w \in W$, $w \in [0, 1]$. W represents a specific threshold set by the service demander, such as the threshold of the quality of service Qos, or the similarity threshold between service operations and requests.

Definition 4: (Composite Service) An abstract composite service ACS can be represented as ACS=($ACINF$, ASS).

(1) ACS: ACS represents the label of the service.

(2) $ACINF$: $ACINF$ represents the feature description information of ACS.

(3) ASS: ASS represents the set of abstract services with successive activation relationships in the composite service flow of ACS. ASS=$\{S_1, S_2, \cdots, S_w\}$, $w = |ASS|$.

Definition 5: (Service Class) Each basic abstract service S corresponds to a specific service class $S(class)$, and the service class can be formally represented as $S(class)$=$\{s_1, s_2, \cdots, s_u\}$. In $S(class)$, $s_1, s_2, \cdots, s_u$ are specific services with the same functional attributes and different non-functional attributes. Use s_i to denote the specific service indexed by i in $S(class)$.

Definition 6: (Specific Composite Service) Each abstract composite service ACS will correspond to a number of specific composite services. A specific composite service CS can be formalized as a 2-tuple $CS = (CINF, SS)$.

(1) CS: CS represents the label of the service.

(2) $CINF$: $CINF$ represents the feature description information of CS.

(3) SS: SS denotes a set of specific services in CS's composite service flow. $SS = \{S_1^{i_1}, S_2^{i_2}, \cdots, S_w^{i_w}\}$, $w = |SS|$. Any $S_t^{i_t} \in SS$ $(1 \leq t \leq w)$ represents a concrete service indexed by i_t in an abstract service S_t.

III. Optimal Semantic Matching Method for Service Bipartite Graph

The essence of local semantic matching is to solve the problem of interface matching between two sub-services. The semantic matching degree between service parameter sets needs to be solved on the basis of the semantic matching degree between the concepts of service parameters. The measurement of the semantic matching degree between service parameter sets can be solved by referring to the optimal matching for bipartite graph [33]. For ease of discussion, the definitions "match service of service request" and "service bipartite graph" are given based on the definition 3 and formulas (1) in [34].

Definition 7: (Match Service of Service Request) Given a service request R=(In^R, Out^R, Pre^R, $Post^R$, W), $w_S, w_{Qos} \in W$, an abstract service S, and a composite service CS, if R and CS satisfy the optimal semantic matching degree[34] $Sim(R, CS) \geq w_S$, $Qos(CS) \geq w_{Qos}$, CS is considered to be a match service of R. In the service database SD, if $\exists CS \in SD$, CS is a match service of R, and $\forall CS_t \in SD, CS \neq CS_t$, satisfy $Sim(R, CS) \geq Sim(R, CS_t) \geq w_S$, $Qos(CS) \geq Qos(CS_t) \geq w_{Qos}$, CS is considered to be an optimal match service of R.

An abstract composite service can be regarded as a composite service demand. The composite demand can be decomposed into a plurality of subtasks, and each subtask corresponds to an abstract service.

Definition 8: (Service Bipartite Graph) Given a composite service $CS \in SD$ and $CS = (CINF, SS)$, for $\forall S_{i-1}, S_i \in SS$, there are two sets of service parameters related to S_{i-1} and S_i, $X = \{x_1, x_2, \cdots, x_n\}$ and $Y = \{y_1, y_2, \cdots, y_m\}$. A service bipartite graph $SG = (X, Y, L, DW)$ is established,, where X and Y are sets of parameter concept vertices, L is a set of edges between concepts, and DW is a set of edges weights. And for $\forall x \in X$, $\forall y \in Y$, $< x, y > \in L$, and $D_W(x, y) \in D_W$, $D_W(x, y)$ is the semantic concept similarity $D(x, y)$[34].

The bipartite graph modeling process is relatively simple and its algorithm complexity is $|L|$ [5]. After a bipartite graph modeled, the calculation of semantic matching degree for X and Y can be converted into the solution of "optimal bipartite matching" $Max(X, Y)$ by a classic "KM" algorithm [5].

The traditional matching method requires that the numbers of nodes in two concept sets are the same when calculating the optimal match. But in fact, the numbers of concepts in two sets may not be the same. Deng Shuiguang et al. [5] extended the optimal matching method for bipartite graph.

The extended optimal matching method can solve the situation where the numbers of concepts in two sets are the same or the number of concepts in the matched concept set is less. However, it ignores the problem where the number of concepts in the matched concept set is more and the matched concept set cannot be completely covered.

Wang Haiyan et al. [33] further extended this method, and the matching method obtained is no longer limited by the number of parameters in concept sets. Referring to the principle of extended optimal matching methods described above, some extended definitions such as "optimal semantic matching for service bipartite graph" are given.

Definition 9: (Extended Service Bipartite Graph) Given a service bipartite graph $SG = (X, Y, L, DW)$, if $|X| = n < |Y|$, then it is necessary to add $l = |Y| - |X|$ virtual vertices to X, the set of virtual vertices is represented as X^+=$\{x_{n+1}, x_{n+2}, \cdots, x_{n+l}\}$. And for each $x^+ \in X^+$, establish $< x^+, y > \in L^+$ for all $y \in Y$for $\forall$ $D_W(x^+, y) \in DW^+$, let $D_W(x^+, y) = 0$, and let $X' = X^+ \cup X$, $L' = L \cup L^+$, $DW' = DW \cup DW^+$, so that an extended service bipartite graph $SG' = (X', Y, L', DW')$ can be obtained. Similarly, if $|X| > |Y|$, an extended service bipartite graph $SG' = (X, Y', L', DW')$ can be obtained, in which $Y' = Y^+ \cup Y$.

Definition 10: (Semantic Matching for Service Bipartite Graph) Given a service bipartite graph $SG = (X, Y, L, DW)$, $M(X, Y)$ is a semantic matching of SG if and only if:

(1) $\forall < x_1, y_1 >, < x_2, y_2 > \in M(X, Y)$, $x_1 \neq x_2$, $y_1 \neq y_2$.

(2) $\bigcup_{\forall x_i, y_i \in M(X,Y)} x_i = X$, $\bigcup_{\forall x_i, y_i \in M(X,Y)} y_i = Y$.

(3) $\forall < x, y > \in M(X, Y)$, $x \leftrightarrow y$.

Definition 11: (Optimal Semantic Matching for Service Bipartite Graph) Given a service bipartite graph $SG = (X, Y, L, DW)$:

(1) If $|X| = |Y|$, $Max(X, Y)$ is an optimal semantic matching for SG if and only if: $Max(X, Y)$ is a semantic matching for SG and for any semantic matching $M(X, Y)$, $Max(X, Y)$ needs to satisfy $\sum_{D_W \in Max(X,Y)} D_W \geq \sum_{D'_W \in M(X,Y)} D'_W$.

(2) If $|X| < |Y|$, an extended service bipartite graph $SG' = (X', Y, L', DW')$ is firstly established, and then $Max(X', Y)$ is an optimal semantic matching for SG, if and only if: $Max(X', Y)$ is a semantic matching for SG, and for any semantic matching $M(X', Y)$, $Max(X', Y)$ needs to satisfy $\sum_{D_W \in Max(X',Y), D_W \neq 0} D_W \geq \sum_{D'_W \in M(X',Y), D'_W \neq 0} D'_W$.

(3) If $|X| > |Y|$, an extended service bipartite graph $SG' = (X, Y', L', DW')$ is firstly established, and then $Max(X, Y')$ is an optimal semantic matching for SG, if and only if: $Max(X, Y')$ is a semantic matching for SG, and for any semantic matching $M(X, Y')$, $Max(X, Y')$ needs to satisfy $\sum_{D_W \in Max(X,Y')} D_W \geq \sum_{D'_W \in M(X,Y')} D'_W$.

Given two sets of concepts X and Y, after the bipartite graph is extended, an example in three different cases is shown in Fig.1. The dashed box in Fig.1 represents the extended service concept vertices. When $|X| = |Y|$, the optimal match after extension is $M(X, Y) = \{< x_1, y_2 >, < x_2, y_1 >, < x_3, y_3 >\}$. When $|X| > |Y|$, the optimal match after extension is $M(X, Y') = < x_1, y_2 >, < x_2, y_1 >, < x_3, y_3 >$. When $|X| < |Y|$, the optimal match after extension is $M(X', Y) = \{ < x_1, y_2 >, < x_2, y_1 >, < x_3, y_3 >\}$, where the extended edge $< x_3, y_3 >$ should be deleted, so the optimal match is $\{< x_1, y_2 >, < x_2, y_1 >\}$.

The related theorems of the extended bipartite graph model can refer to Theorem 3 and Theorem 4 in paper [5].

Definition 12: (Parameter Dependency Relationship between Services) Given a set of service operations FUN, for $\forall$ Fun_i, $Fun_j \in FUN$, if $\exists$ $out_s \in Out_i$, $\exists in_s \in In_j$, satisfy $out_s \leftrightarrow in_s$, then there is a parameter dependency relationship between Fun_i and Fun_j, expressed as $Fun_i \xrightarrow{OI} Fun_j$. $Max(Out_i, In_j)$ is used to represent the optimal bipartite matching for Out_i and In_j.

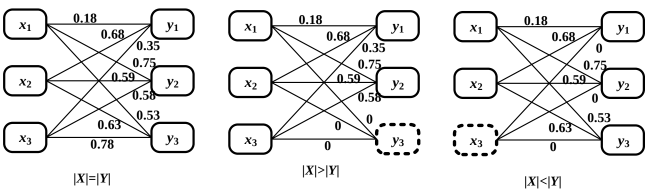


Fig. 1. The Extended Best Match in Three Cases

Definition 13: (Condition Dependency Relationship between Services) Given a set of service operations FUN, for $\forall\ Fun_i, Fun_j \in FUN$, if $\exists\ post_s \in Post_i$, $\exists pre_s \in Pre_j$, satisfy $post_s \leftrightarrow pre_s$, then there is a condition dependency relationship between Fun_i and Fun_j, expressed as $Fun_i \xrightarrow{Con} Fun_j$. In this paper, $post_s$ and pre_s can be expressed as RDF condition triples. $Max(Post_i, Pre_j)$ is used to represent the optimal bipartite matching for $Post_i$ and Pre_j.

Given $|Out_{i-1}| = |In_i| = 3$, if X and Y in Fig.1 are replaced with Out_{i-1} and In_i respectively, the optimal matching bipartite graph between Out_{i-1} and In_i can be obtained based on the definition 11 and the KM algorithm.

Based on the above extended bipartite graph model and related definitions, the formulas for the matching degree between parameter sets of S_{i-1} and S_i are obtained. (a): If there is no successive activation relationship between S_{i-1} and S_i. (b): If there is a successive activation relationship between S_{i-1} and S_i.

$$SIMD_{IO}(S_{i-1}, S_i) = \begin{cases} 0, & (a) \\ \dfrac{\sum_{D_W \in Max(Out_{i-1}, In_i)} D_W}{|Max(Out_{i-1}, In_i)|}, & (b) \end{cases} \tag{1}$$

$$SIMD_{Con}(S_{i-1}, S_i) = \begin{cases} 0, & (a) \\ \dfrac{\sum_{D_W \in Max(Post_{i-1}, Pre_i)} D_W}{|Max(Post_{i-1}, Pre_i)|}, & (b) \end{cases} \tag{2}$$

In the above formulas, $SIMD_{IO}(S_{i-1}, S_i)$ represents the matching degree between Out_{i-1} of S_{i-1} and In_i of S_i, $SIMD_{Con}(S_{i-1}, S_i)$ represents the matching degree between $Post_{i-1}$ of S_{i-1} and Pre_i of S_i.

Theorem 1: Given any composite service $CS \in SD$ and $CS = (CINF, SS)$, for $\forall\ S_{i-1}$, $S_i \in SS$, the following properties hold:

(1) $SIMD_{IO}(S_{i-1}, S_i) \in [0, 1]$.

(2) $SIMD_{Con}(S_{i-1}, S_i) \in [0, 1]$.

Proof: It is easy to draw theorem 1 based on definition 8.

Based on the above analysis, Algorithm 1 (Service Parameter Semantic Matching Algorithm, abbreviated as SPSM) is proposed.

Algorithm 1: $SPSM$

Input: Service related concept sets X and Y.
Output: $Max(X, Y)$ and *SIMD*.
for $\forall x \in X$ **do**
  for $\forall y \in Y$ **do**
    $< x, y > \to L$;
    Based on formulas (1) to (3) in [34], gets $D_W(x, y) = D(x, y)$, $D_W(x, y) \to DW$;
  end
end
Gets the service parameter bipartite graph $SG = (X, Y, L, DW)$;
Based on definition 11 and KM algorithm, gets the optimal semantic matching $Max(X, Y)$;
for $\forall D_W \in Max(X, Y)$ **do**
  $D_W \to SD_W$, $ND_W + +$;
end
$SIMD = SD_W / ND_W$;

IV. Quality Evaluation Model based on Semantic Matching Degree and Composite Service Quality

After solving the problem of local semantic matching, a reasonable quality evaluation model for composite services needs to be designed to solve the problem of global semantic matching between requirements and composite services.

There are four main control structures involved in service composition process, including sequential structure, selective structure, parallel structure and cyclic structure, as shown in Fig.2. Because some technical methods (such as Critical Path Algorithm, Loop Unfolding Algorithm, etc.) can be used to convert other structures into sequential structures, some papers only consider the sequential structure [35].

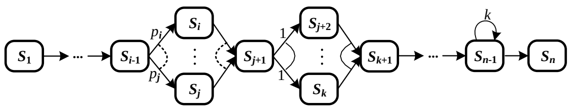


Fig. 2. Control Structures Involved in Service Composition Process

The Qos non-functional attributes mainly include: execution cost, execution time, availability, and reliability. In a QoS quality model, different non-functional attributes may have different dimensions and dimension units, and sometimes may differ by several orders of magnitude. The magnitude difference will inevitably affect the aggregation and data analysis of the Qos attribute values in a composite service, thus affecting the validity of the quality model. In order to eliminate the influence of dimensions, the data can be first normalized according to Table I, so that each data is in the same order of magnitude. Then, according to Table II, the combined data of each Qos attribute is obtained, and then the QoS comprehensive quality model is obtained and evaluated.

At present, most papers are compared with the Service Matchmaking algorithm [5] when analyzing performance for dynamic service composition and selecting algorithms. The algorithm considers the optimal semantic matching between user requirements and the input/output parameters of service operations, and considers the dependency relationship between the output and input parameters of service operations.

However, the problem of interface and condition semantic matching between multiple sub-services in a composite service is not considered, so that the correctness of the data flow between sub-services cannot be guaranteed. Therefore, the Service Matchmaking algorithm [5] only considers “global semantic matching” and does not consider the correctness and validity of composite services based on “local semantic matching”, and this is exactly the starting point of this paper.

At the same time, compared with other related papers

TABLE I
NORMALIZED PROCESSING FUNCTIONS FOR EACH Qos ATTRIBUTE

Qos Attributes	Normalized processing functions F_{sp}
Cost	$F_{sp}(Qos_{cost}(CS)) = 1/Qos_{cost}(CS)$
Time	$F_{sp}(Qos_{time}(CS)) = 1/Qos_{time}(CS)$
Availability	$F_{sp}(Qos_{avl}(CS)) = ln(Qos_{avl}(CS))$
Reliability	$F_{sp}(Qos_{rel}(CS)) = ln(Qos_{rel}(CS))$

[5,22-32,34], it is found that: (1) although the paper [33] considers the data stream transmission among sub-services within a composite service and considers the semantic matching for condition parameters, it does not consider the dependency relationship between the input and output parameters of a service (operation); (2) although the paper [5] considers the dependency relationship between the input and output parameters of a service (operation), it does not consider the data stream transmission among sub-services within a composite service. And the paper [5] only considers the global parameter matching degree between the request R and the service S from the perspective of the input and output parameter dependence, and does not consider the impact of the input and output parameter dependency on the validity of composite service; (3) the method for parameter semantic matching degree proposed in [25,30,36,38] does not consider whether input parameters of a subsequent service are completely covered or not; (4) the paper [37] considers the local semantic matching of services, but does not consider the impact of condition dependency in composite services, so there is a problem that the global efficiency of composite services is low.

In order to solve the above problems, this paper will take into account the local and global semantic matching to design a quality evaluation model to cope with the problem of dynamic service composition and selection.

The Web Services Registration Model(WSRM) [5] supports the semantic tagging of service interfaces and the declaration of interface dependencies. The following discussion is based on this model, and definitions 14 to 15 are proposed in conjunction with relevant definitions and examples in [5].

Definition 14: (The Input Parameter Set Dependent on by an Output Parameter). Given a function operation $Fun = (Inf, In, Out, F_{IO}, Pre, Post, Qos)$, for $\forall\ out_t \in Out$, satisfy $F_{IO}(out_t) \subseteq In$. For $\forall\ in \in In$, if the output parameter out_t depends on the input parameter in, then in is called the input parameter that out_t depends on, and $F_{IO}(out_t)$ is called the input parameter set that out_t depends on.

Definition 15: (Parameter Dependency Degree of an Output Parameter) Given a service operation $Fun = (Inf, In, Out, F_{IO}, Pre, Post, Qos)$, for $\forall\ out_t \in Out$, satisfy $F_{IO}(out_t) \subseteq In$. The closeness degree of the output interface parameter out_t depending on the input parameter set In can be expressed as $DF(out_t) = |F_{IO}(out_t)|/|In|$. If $DF(out_t) = 1$, out_t is said to be dependent on In, expressed as $out_t \to In$. If $DF(out_t) = k < 1$, out_t is said to be partially depends on In, expressed as $out_t \xrightarrow{k} In$.

Based on the above formulas, definitions, and analysis, a quality evaluation model $Qos(CS)$ based on semantic matching and composite service quality is proposed:

$$Qos(CS) = W_A * Qos_A(CS) + W_B * Qos_B(CS) + W_C * Qos_C(CS) + W_D * Qos_D(CS) \quad (3)$$

Where,

$$Qos_A(CS) = \alpha * Qos_{IO}(CS) + \beta * Qos_{con}(CS) \quad (4)$$

$$Qos_B(CS) = \prod_{i=|CS|}^{1} SIMD(Out_{i-1}, \bigcup_{out_t \in Out_i} F_{IO}(out_t)) \quad (5)$$

$$Qos_C(CS) = \sum_{out_t \in Out^R} [SIMD(out_t, MaxF(R, CS, out_t)) * \prod_{in_s \in F_{IO}(MaxF(R,CS,out_t))} SIMD(in_s, MaxF(R, CS, in_s))]/|Out^R| \quad (6)$$

$$Qos_D(CS) = \sum_{k=1}^{4} w_k * Qos_k(CS) \quad (7)$$

$$MaxF(R, CS, out_t) = \{out_s|(out_t, out_s) \in Max(Out^R, Out^{CS})\} \quad (8)$$

$$Qos_{IO}(CS) = (\sum_{S_i \in CS} SIMD_{IO}(S_{i-1}, S_i))/|CS| \quad (9)$$

$$Qos_{con}(CS) = (\sum_{S_i \in CS} SIMD_{con}(S_{i-1}, S_i))/|CS| \quad (10)$$

CS is a composite service. The weight coefficients satisfy $W_A + W_B + W_C + W_D = 1$ and $\sum_{k=1}^{4} w_k = 1$. $Qos_A(CS)$ measures the global matching degree of the interface parameters and condition parameters between the sub-services in a composite service. It evaluates the delivery of data flows between sub-services to ensure seamless connectivity between sub-services of a composite service. $Qos_B(CS)$ considers the influence of the dependency between the output and input parameters on the validity of a composite service. $Qos_B(CS)$ starts from the last sub-service, for $\forall\ out_t \in Out_i$ finds $F_{IO}(out_t)$ and obtains $\bigcup_{out_t \in Out_i} F_{IO}(out_t)$. Based on this, obtains $SIMD(Out_{i-1}, \bigcup_{out_t \in Out_i} F_{IO}(out_t))$. Considering all matching degree synthetically, it obtains the dependency of the output parameters on the input parameters in a composite service. The higher the dependency degree is, the more reasonable the composite service is. $Qos_C(CS)$ considers the global parameter matching degree between a request R and a service CS from the perspective of the input and output parameter dependence. $Qos_D(CS)$ is CS's QoS quality model.

Based on the above analysis, the following Algorithm 2 are proposed.

Theorem 2: For any one composite service $CS = (CINF, SS)$, the following properties hold:

(1) $Qos_A(CS) \in [0, 1]$.
(2) $Qos_B(CS) \in [0, 1]$.
(3) $Qos_C(CS) \in [0, 1]$.
(4) $Qos_D(CS) \in [0, 1]$.

Proof: (1) From Theorem 1, it can be seen that $SIMD_{IO}(S_{i-1}, S_i) \in [0, 1]$ and $SIMD_{Con}(S_{i-1}, S_i) \in [0, 1]$. Therefore, $Qos_{IO}(CS) \in [0, 1]$ and $Qos_{Con}(CS) \in$

TABLE II
VARIOUS QOS ATTRIBUTE COMBINATION RULES FOR DIFFERENT CONTROL STRUCTURES

Qos Attributes	Order Structure	Selection Structure	Parallelism Structure	Loop Structure
Cost	$\sum_{s_i \in CS} Qos_{cost}(S_i)$	$\sum_{s_i \in CS} Qos_{cost}(S_i) * p_i$	$\sum_{s_i \in CS} Qos_{cost}(S_i)$	$k * Qos_{cost}(S_i)$
Time	$\sum_{s_i \in CS} Qos_{time}(S_i)$	$\sum_{s_i \in CS} Qos_{time}(S_i) * p_i$	$Max_{s_i \in CS}(Qos_{time}(S_i))$	$k * Qos_{time}(S_i)$
Availability	$\prod_{s_i \in CS} Qos_{avl}(S_i)$	$\prod_{s_i \in CS} Qos_{avl}(S_i) * p_i$	$Min_{s_i \in CS}(Qos_{avl}(S_i))$	$(Qos_{avl}(S_i)^k)$
Reliability	$\prod_{s_i \in CS} Qos_{rel}(S_i)$	$\prod_{s_i \in CS} Qos_{rel}(S_i) * p_i$	$Min_{s_i \in CS}(Qos_{rel}(S_i))$	$(Qos_{rel}(S_i)^k)$

Algorithm 2: Qos

Input: A composite service $CS = (CINF, SS)$, weight coefficients α, β, W_A, W_B, W_C and W_D.
Output: $Qos(CS)$.

for $\forall S_{i-1},\ S_i \in SS$ **do**
  Call algorithm 1 to get $SIMD_{IO}(S_{i-1}, S_i)$ and $SIMD_{Con}(S_{i-1}, S_i)$;
  $SIMD_{IO}(S_{i-1}, S_i) \rightarrow SUM_{IO}$;
  $SIMD_{Con}(S_{i-1}, S_i) \rightarrow SUM_{Con}$;
end
Based on formulas (4), (9) and (10), get $Qos_A(CS)$.
for $\forall S_{i-1}, S_i \in SS$ **do**
  for $\forall out_t \in Out_i$ **do**
    $F_{IO}(out_t) \rightarrow In_i^{F_{IO}}$;
  end
  Call algorithm 1 to get $SIMD(Out_{i-1}, In_i^{F_{IO}})$;
end
Based on formulas (5), get $Qos_B(CS)$;
for $\forall out_t \in Out^R$ **do**
  Call algorithm 1 to get $Max(Out^R,\ Out^S)$;
  Get $MaxF(R, CS, out_t)$ by the formula (8);
  Call algorithm 1 to get $SIMD_A(out_t, MaxF(R, CS, out_t))$;
  for $\forall out_s \in MaxF(R, S, out_t)$ **do**
    $F_{IO}(out_s) \rightarrow In_s^{F_{IO}}$;
  end
  for $\forall in_s \in In_s^{F_{IO}}$ **do**
    Call algorithm 1 to get $SIMD(in_s, MaxF(R, CS, in_s))$;
    $SIMD_B = SIMD_B * SIMD$;
  end
  $SIMD_{AB} = SIMD_A * SIMD_B$;
  $Qos_C(CS) = Qos_C(CS) + SIMD_{AB}$;
end
$Qos_C(CS) = Qos_C(CS)/|Out^R|$;
According to the service quality method for composite services (Table I and Table II), obtains $Qos_D(CS)$
Get $Qos(CS)$ by the formula (3).

$[0, 1]$. Since weight coefficient α and β satisfy $\alpha + \beta = 1$, $Qos_A(CS) \in [0, 1]$ is obtained.

(2) Since $\bigcup_{out_t \in Out_i} F_{IO}\ (out_t) \subseteq In_i$, $SIMD(Out_{i-1},$ $\bigcup_{out_t \in Out_i} F_{IO}\ (out_t)) \in [0, 1]$ can be known from Theorem 1. Therefore, $Qos_B(CS) \in [0, 1]$.

(3) From Theorem 1, it can be seen that $SIMD(in_s,$ $MaxF(R,\ CS,\ in_s)) \in [0, 1]$. Then get $(\prod_{in_s \in F_{IO}(MaxF(R,CS,out_t))}\ SIMD(in_s,\ MaxF(R,\ CS,$ $in_s)))\ *\ SIMD\ (out_t,\ MaxF(R,\ CS,\ out_t)) \in [0, 1]$.

Such that, $\sum_{out_t \in Out^R}\ [SIMD\ (out_t,\ MaxF(R,$ $CS,\ out_t))\ *\ \prod_{in_s \in F_{IO}(MaxF(R,CS,out_t))}\ SIMD(in_s,$ $MaxF(R,\ CS,\ in_s))] \in [0, |Out^R|]$. Therefore, $Qos_C(CS) \in [0, 1]$ is obtained.

(4) Easy to know, $1 \leq k \leq 4$, $Qos_k(CS) \in [0, 1]$, weight coefficients satisfy $\sum_{k=1}^{4} w_k = 1$. Therefore, $Qos_D(CS) \in$ $[0, 1]$.

Theorem 3: Given a composite service $CS \in SD$ and $CS = (CINF, SS)$, $Qos(CS) \in [0, 1]$ holds.

Proof: It is easy to prove by Theorem 2.

At the same time, inspired by the definitions "the pareto solution" [30] and "the skyline service" [39], the definitions "Qos-dominant relationship between services" and "set of Qos high-quality solutions for services" are proposed.

Definition 16: (Qos-dominant Relationship between Services) For $\forall\ CS_i, CS_j \in SD$, each service has Qos attributes including Qos_A, Qos_B, Qos_C and Qos_D. For any one Qos attribute $Qos_k(CS) \in \{Qos_A,\ Qos_B,\ Qos_C,\ Qos_D\}$ (see formulas (4) to (7)), there exits $Qos_k(CS_i) \geq Qos_k(CS_j)$. And $\exists\ Qos_t(CS_i) \in \{Qos_A, Qos_B, Qos_C, Qos_D\}$, satisfies $Qos_t\ (CS_i) > Qos_t(CS_j)$, then CS_i is Qos-dominated CS_j, expressed as $CS_i >_{\ Qos} CS_j$.

Definition 17: (Set of Qos High-quality Solutions for Services) For $\forall\ CS_i \in SD$, if $\nexists\ CS_j \in SD$ satisfies $CS_j >_{Qos}\ CS_i$, then the set of all the service that is not Qos-dominant by other services is called the set of Qos high-quality solutions. The set of Qos high-quality solutions can be formally expressed as: $HQS = \{CS_i \in SD \mid \nexists\ CS_j \in SD,\ CS_j >_{Qos} CS_i\}$.

V. INTELLIGENT COMBINATION AND SELECTION ALGORITHM FOR SEMANTIC WEB OF THINGS SERVICES

Multi-objectives and multi-constraints problems need to resolve the conflict between multiple objectives. It requires a trade-off between multiple objectives to obtain a set of high-quality solutions satisfying the constraints [22]. The Qos-based dynamic service composition and selection problem is a multi-objectives and multi-constraints problem and has been proved to be a NP-Complete problem [40]. Genetic algorithm has many characteristics, such as parallel, intelligent, global convergence and linear time complexity, and is suitable for solving such problems [41]. Compared with the traditional multi-objectives optimization method, only the objective function and fitness function of genetic algorithm need to be set without providing any other heuristic knowledge related to the application background of the problem [22].

At present, there are many related papers on service composition and selection based on genetic algorithm [21-31,40]. It is found that the use of genome, gene bit and multiple operators in genetic algorithm can be a good representation of the local and global semantic matching. Therefore, this paper will design a dynamic service composition and selection algorithm based on the multi-objectives and multi-constraints genetic algorithm.

The algorithm encodes a service composition process as a chromosome, generates new chromosomes more in line with user requirements through crossover, mutation, and selection operations between chromosomes. This iterative process is continuously performed to achieve parallel global search in the solution space. When the algorithm stops, chromosomes those meets user's needs are obtained. The relevant design steps of the algorithm are as follows.

Coding designs usually use integer coding. A composite service can be encoded as a genome (chromosome) and an abstract sub-service in the composite service can be represented as a gene bit. The range of the gene bit is related to the candidate set of specific services for the corresponding abstract sub-service. A decimal number can be used to represent the serial number of a specific service in a service class and correspond to a gene bit. An example of genome is shown in Fig.3.

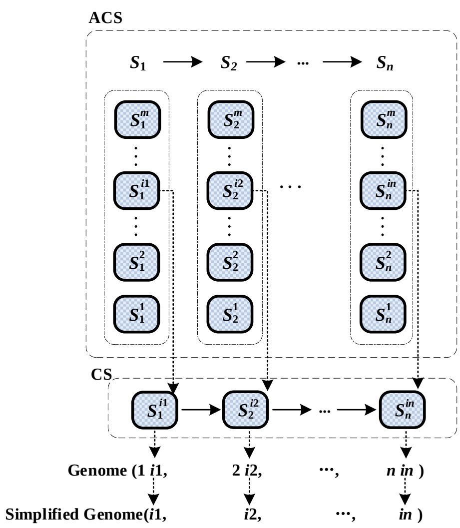


Fig. 3. Genomic Coding Design

Suppose an abstract composite service ACS consists of n abstract services S_1, S_2, $\cdots$, S_n. Each abstract service S_i corresponds to a service class, and each service class contains m candidate specific services $\{S_i^1, S_i^2, \cdots, S_i^m\}$. In a specific composite service $\{S_1^{i1}, S_2^{i2}, \cdots, S_n^{in}\}$, S_t^{it} represents a specific service for S_t with the serial number it. Since a genome consists of n gene bits and the genome length is fixed, service class numbers can be omitted, and only one decimal number is used to indicate the serial number of a specific service in it's service class, and correspond to one gene bit.

$Qos(CS)$ in formula (3) can be defined as an objective function. Then, the fitness function $Fitness(CS)$ can be constructed by using a penalty function method to integrate the global limit with the objective function $Qos(CS)$. $Fitness(CS)$ is used to calculate the fitness of each genome in the population, and the genome is selected according to it's fitness. The higher the fitness is, the greater the chance that the genome participates in constructing a new population is.

Combined with roulette and the optimal individual preservation method, each generation of Qos high-quality solutions is stored in a set of high-quality solutions. And then the individuals are selected for crossover and variation by roulette method. Adopt the multi-parent and two-point crossing method. Firstly, it is necessary to determine two crossover positions in multiple parent individuals, and then exchange gene strands between the two crossover positions among parent individuals. The crossover positions can be selected based on Qos, as shown in Fig.4. The mutations are randomly disturbed and randomly mutated according to mutation probability. A gene is randomly selected (ie, an abstract service in the service composition process), and a candidate service corresponding to the gene is randomly used to replace the current specific service, as shown in Fig.5.

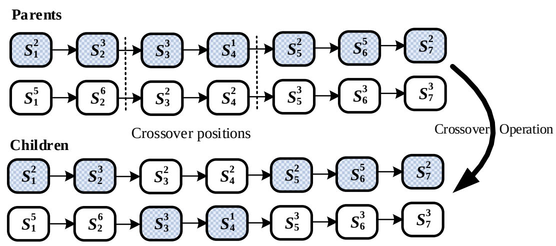


Fig. 4. Two-point Crossover Operation

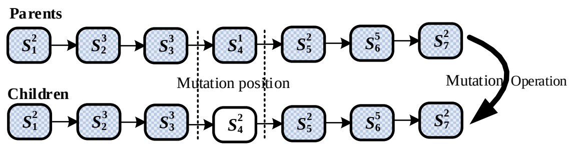


Fig. 5. Mutation Operation

Three stop conditions in the process of evolution are set. Stop condition 1, set the maximum number of evolutionary generation in advance, if the number of generation is larger than the maximum, the evolution is terminated. Stop condition 2, when a new generation of population is generated, the difference between the mean value of fitness functions in the new population and the mean value of fitness functions in the previous generation is calculated to determine whether the difference is less than the threshold value. If the difference is less than the threshold value, it means the fitness function of the new generation do not change much, and the evolution will be terminated [42]. Stop condition 3, calculate the mean square deviation of the fitness functions in the new population, which represents the diversity and evolution degree of the population. When the mean square deviation is greater than the threshold, the iteration is stopped [43]. If one of the three conditions is satisfied, the algorithm is stopped, and the set of Qos high-quality solutions in the current population is output, or after the solutions are sorted in conjunction with the user context ($Sort(SP')$ in step 28 in algorithm 3), the sorted results is recommended to users.

Different from the fitness function of traditional genetic algorithms, the objective function of the fitness function proposed in this paper is composed of four parts, including $Qos_A(CS)$, $Qos_B(CS)$, $Qos_C(CS)$ and $Qos_D(CS)$. Considering the heterogeneity and priority of each part, this paper sets weight for each part. The fitness value of each chromosome is obtained from these four parts and the penalty coefficient, which respectively represent the chromosome adaptability from different angles. Refer to formula (11) for the definition of the fitness function.

$$Fitness(CS) = Qos(CS) - \tau \sum_{k=1}^{4} (\triangle Q_k/(Q_k^{max} - Q_k^{min}))^2 \quad (11)$$

Q_k^{max} represents the maximum value in the kth constraint. Q_k^{min} represents the smallest value in the kth constraint. There are four constraints, namely, execution cost, execution time, availability and reliability. Q_k represents the value of operations for some constraints or the value of a related constraint on service CS. $\triangle Q_k$ [21] is defined as follows.

$$\triangle Q_k = \begin{cases} Q_k - Q_K^{max}, & Q_k^{max} < Q_k \\ 0, & Q_k^{min} \le Q_k \le Q_k^{max} \\ Q_k^{min} - Q_k, & Q_k < Q_k^{min} \end{cases} \quad (12)$$

Theorem 4: Given a composite service $CS \in SD$ and $CS = (CINF, SS)$, CS satisfies $Fitness(CS) \in [0, 1]$.

Proof: The proof is easily derived by Theorem 3 and standard deviation method [44].

Mutation operator is used as auxiliary operator because of its local search ability, and the mutation probability is generally lower. Crossover operator is used as the main operator because of its global search ability. With the evolution of population, the diversity of population will continue to improve. In order to avoid the occurrence of premature convergence, the crossover probability and mutation probability should be higher in the initial stages and lower in the later stages [45]. Therefore, in this paper, the mean square deviation $MSE(SP)$ of fitness functions for the population SP is used to set the coefficient γ of crossover probability cp and mutation probability mp as follows.

$$Average\_F(P) = \sum_{p \in P} Fitness(p)/|P| \quad (13)$$

$$MSE(P) = \sum_{p \in P} (Fitness(p) - Average\_F(P))^2/|P| \quad (14)$$

$$\gamma = 1 - MSE(P) * g/g_{max} \quad (15)$$

Based on the above analysis, an intelligent combination and selection algorithm for Semantic Web of Things services (abbreviated as IC&S_SWTS) is proposed, as shown in Algorithm 3.

Algorithm 3: $IC\&S\_SWTS$

Input: Service Population SP,g_{max},cp_{max},mp_{max}.
Output: Sorted solutions *Result*.

```
Initialize population SP, |SP| = N;
cp_max → cp; mp_max → mp;
for ∀CS ∈ SP do
    Call algorithm 2 to calculate the objective function Qos(CS);
    Calculate Fitness function Fitness(CS) by formula (11);
end
g = 1;
while g ≤ g_max do
    ∅ → Temp; ∅ → SP';
    Select_HQS(SP) → SP' ;
    Select(SP − SP', |Temp| − |SP'|) ∪ SP' → Temp;
    for ∀cs_1, cs_2, ··· , cs_t ∈ Temp do
        cross_index = Max_Qos(cs_1, cs_2, ··· , cs_t);
        Crossover(cs_1, cs_2, ···, cs_t, cross_index) → Temp;
    end
    Mutation(Temp, mp) → Temp;
    Select_HQS(SP ∪ Temp) → SP';
    SP = SP' ∪ Select(SP − SP', N − |SP'|);
    Average_F(SP') → average_f[g];
    if average_f[g] - average_f[g − 1]) ≤ θ_1 or MSE(SP) ∗ g/gmax ≥ θ_2 then
        break;
    else
        γ = 1 − MSE(SP) ∗ g/g_max;
        γ ∗ cp → cp; γ ∗ mp → mp;
    end
    g++;
end
Sort(SP');
```

VI. Experiment Analysis

The datasets EEE-05 and ICEBE-05 are used primarily for service discovery and service composition competitions. This paper selects the Composition Dataset from the dataset ICEBE-05. The Composition Dataset contains 26,904 WSDL files for web services, 100 of which were selected in this paper. Based on the reference and call relationships between these services, relevant networks are formed and tested. In the service dataset, each QoS parameter for each service is assigned randomly. All attribute information (including interface parameters, etc.) of the services are semantically annotated using the domain ontology concept tree selected from the WordNet concept library. In the same way, service demands are also semantically annotated using domain ontology sources [46-47]. In order to verify the feasibility and validity of the method proposed in this paper, experiments select Intel Core 1.00GHz, 4.00GB RAM and Windows XP environment to conduct comparative experiments on algorithms.

The main factors affecting the performance of service algorithms include the number of service classes (number of tasks) in composite services, the number of candidate services in each service class, and the fitness function.

The number of service class n is set to 6. The number of candidate services in each service class is m, $m \in [10, 100]$. The size of service population N is 50. With the changes of m, the maximum number of evolutionary generations $gmax$ is adjusted accordingly, $gmax \in [100, 400]$. The maximum crossover probability cp_{max} is 1 and the maximum mutation probability mp_{max} is 0.01. The weight coefficients in formula (3) are $W_A = W_B = W_C = 0.2$ and $W_D = 0.4$, respectively. The weight coefficients α and β in formula (4) are both set to 0.5. For $\forall\ k \in [1,4]$, the weight coefficient w_k in formula (7) is set to 0.25. The dynamic service composition and selection problem is transformed into multi-objective and multi-constrained optimization problem based on formula (11) and genetic algorithm. At the same time, within the range of quality parameters, four quality of service parameters are randomly generated for each candidate service.

Algorithms $SM-based$ [33] and $ServiceMatchmaking$ [5] are the starting point of this paper. Algorithm $ServiceMatchmaking$ considers the dependency relationship between the output and the input parameters of services. It focuses on "global semantic matching" and does not consider the correctness and validity of composite services based on "local semantic matching". And the algorithm is not implemented using genetic algorithm. Algorithm $SM-based$ considers "local semantic matching", takes into account the semantic matching of interface parameters and the semantic matching of condition parameters, and is solved by genetic algorithm. At the same time, this paper also selects the $CQCA$ algorithm [20] for comparative experiments. $CQCA$ algorithm proposed by Khanouche M E is an intelligent service composition algorithm based on clustering and QoS perception. The algorithm divides the candidate services into multiple clusters, where each cluster corresponds to a QoS level. It uses the utility function to filter candidate services, and constructs a search tree to find approximate optimal combinations.

Algorithm $IC\&S\_SWTS$ designed in this paper takes into account "global semantic matching" and "local semantic matching". In it's objective function, $Qos_A(CS)$ evaluates data flows between sub-services to ensure seamless connection between sub-services in a composite service [33], $Qos_C(CS)$ considers the global parameter matching degree between a request R and a service CS [5], and $Qos_D(CS)$ is a QoS quality model for composite services. $Qos_B(CS)$ is designed based on papers [5,34]. It uses the dependency function F_{IO} to represent $SIMD(Out_{i-1}, \bigcup_{out_t \in Out_i} F_{IO}(out_t))$. It obtains the total dependence degree of output parameters on input parameters among all the sub-services in a composite service, as shown in formula (5). The higher the $Qos_B(CS)$ is, the more validity CS is. Specific definitions can refer to formulas (4) to (7).

Dynamic service composition and selection algorithms should be able to dynamically and timely respond to service requirements and provide more accurate services. Therefore, time performance and result accuracy are tested in this paper.

It can be seen from Fig.6.(a) and Fig.6.(b) that the time performance of the algorithm $CQCA$ [20], the algorithm $SM-based$ [33], the algorithm $ServiceMatchmaking$ [5] and the new algorithm $IC\&S\_SWTS$ are similar. Among them, the fitness function of the algorithm $IC\&S\_SWTS$ is the formula (11). As the number of candidate services changes, the time consumption of $IC\&S\_SWTS$ is slightly higher, but the time curves of the four algorithms are basically linearly increasing and are relatively close.

The maximum time difference between the algorithm $IC\&S\_SWTS$ and the algorithm $SM-based$ is 0.8 seconds, the maximum time difference between the algorithm $IC\&S\_SWTS$ and the algorithm $ServiceMatchmaking$ is less than 1.1 seconds, and the maximum time difference between the algorithm $IC\&S\_SWTS$ and the algorithm $CQCA$ is less than 1.6 seconds. These results show that although $Qos_B(CS)$ and $Qos_C(CS)$ have been newly added in $IC\&S\_SWTS$, these two assessment factors have less impact on time performance and their time consumption is negligible.

Fig.6.(a) and Fig.6.(b) show that as the number of candidate services increases and the maximum number of evolutionary generations changes, $IC\&S\_SWTS$'s runtime does not change drastically. This shows that the preset problem size and execution time can basically meet most of requirements.

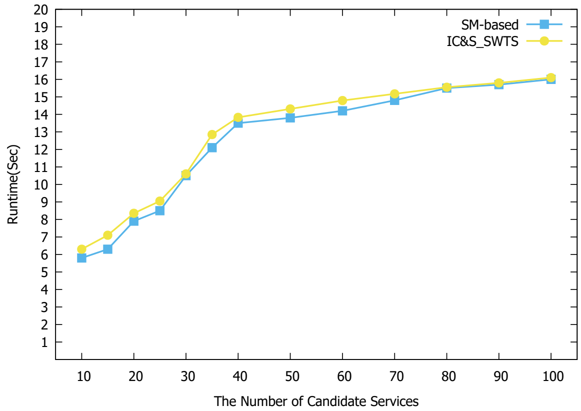
In Fig.6.(c), the fitness function of $IC\&S\_SWTS'$ contains only $Qos_A(CS)$, $Qos_C(CS)$ and $Qos_D(CS)$, not including $Qos_B(CS)$. When testing $IC\&S\_SWTS'$, the weights in formula (3) need to be adjusted to $W_A = W_C = 0.3$, $W_B = 0$, and $W_D = 0.4$. As can be seen from Fig.6.(c), the assessment factor $Qos_B(CS)$ has little effect on $IC\&S\_SWTS$'s runtime.

Fig.6.(d), Fig.6.(e), and Fig.6.(f) analyze the precision of algorithms. The precision is defined as follows: the precision ratio of service = (the number of services that can actually meet user's needs)/(the number of returned services that satisfy user's needs). The precision of each algorithm is calculated.

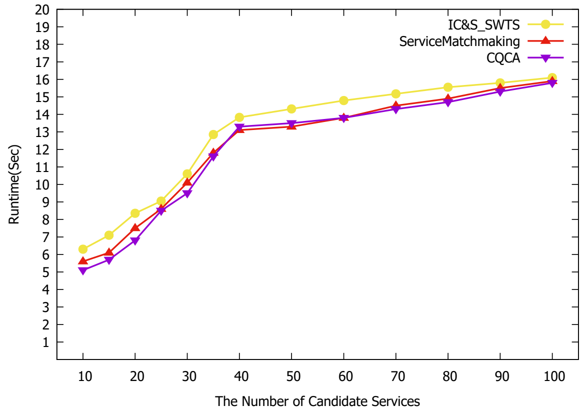
The comparison of the algorithm $IC\&S\_SWTS$ with the other three algorithms is shown in Fig.6.(d). Through comparison and analysis, it is found that the algorithm $ServiceMatchmaking$ focuses on "global semantic matching", does not consider the correctness and validity of composite services based on "local semantic matching", and its precision is slightly lower. Since $SM-based$ does not consider the dependency relationship between input and output parameters in an operation, its validity and precision is also slightly lower. The $CQCA$ algorithm can shorten the time to a certain extent, but it does not consider the correctness and validity of composite services based on "local semantic matching", resulting in a slightly lower effectiveness of the composite service. The algorithm $IC\&S\_SWTS$ has the highest precision.

In Fig.6.(e), the second algorithm is $IC\&S\_SWTS'$. The fitness function in $IC\&S\_SWTS'$ does not include $Qos_B(CS)$. $IC\&S\_SWTS'$ is used to test the effect of $Qos_B(CS)$ on the validity of results. It can be seen that the precision of $IC\&S\_SWTS'$ is lower. To further measure the effect of $Qos_B(CS)$ on the validity, weight coefficients in formula (3) need to be adjusted to three cases, namely: (a) $W_A = W_B = W_C = 0.2$, $W_D = 0.4$; (b) $W_A = W_C = 0.22$, $W_B = 0.16$, $W_D = 0.4$; (c) $W_A = W_C = 0.25$, $W_B = 0.1$, $W_D = 0.4$. In Fig.6.(f), the three cases correspond to $IC\&S\_SWTS$, $IC\&S\_SWTS_1$ and $IC\&S\_SWTS_2$. It can be seen that as the weight coefficient of Qos_B(CS) decreases, the precision decreases.

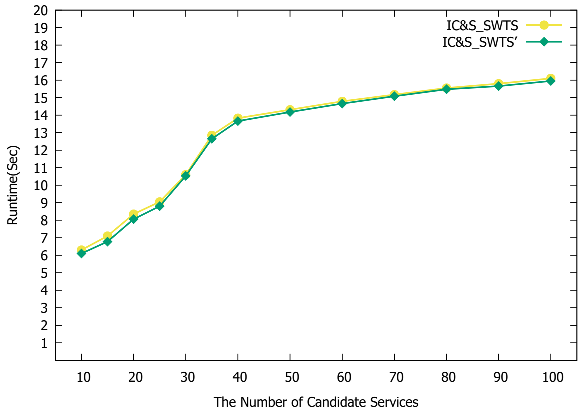
After analysis, it is found that $Qos_B(CS)$ has a large impact on the precision of results. The fitness function model and $IC\&S\_SWTS$ proposed in this paper can obtain better composite service results. The ratio of approximate optimal solutions obtained by $IC\&S\_SWTS$ reaches more than 95%, which is higher than other algorithms. In short, $IC\&S\_SWTS$ can provide reasonable approximate optimal



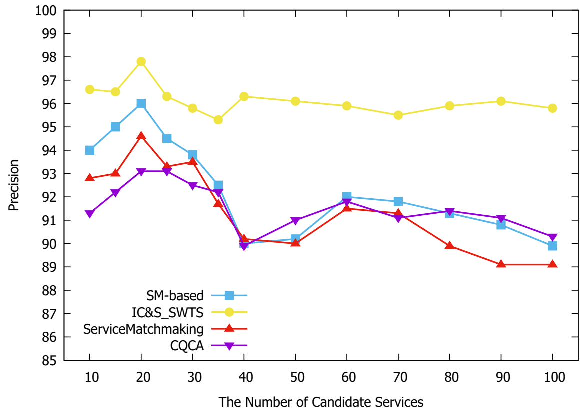
(a) The Time Performance between $SM-based$ and $IC\&S\_SWTS$



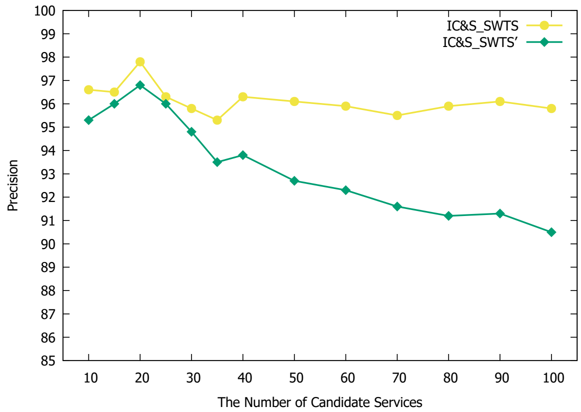
(b) The Time Performance of Three Algorithms



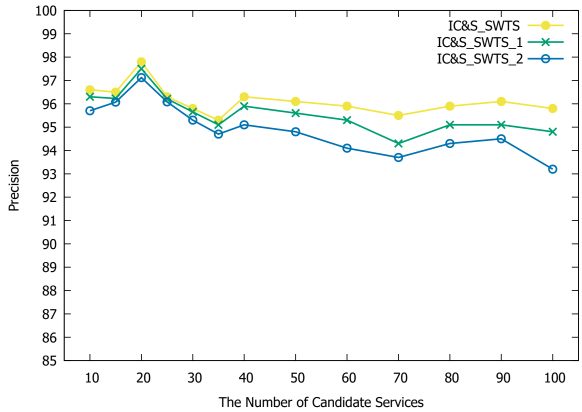
(c) Comparative Analysis of the Influence of $Qos_B(CS)$ on Runtime



(d) The Precision of Four Algorithms



(e) The Precision between $IC\&S\_SWTS$ and $IC\&S\_SWTS'$



(f) Comparative Analysis of the Influence of $Qos_B(CS)$ on Precision

Fig. 6. Experiment Analysis

solutions on the premise of lower cost.

VII. Conclusion

As a kind of intelligent Internet of Things, SWoT should be able to provide efficient and dynamic services for users in a timely manner. However, the existing static Web services composition methods basically cannot meet the real-time and dynamic requirements, and cannot realize the matching and discovery of dynamic service resources. Therefore, the research on dynamic service composition and selection is particularly important. At present, the Qos-based dynamic service composition and selection problem is an important research direction in SWoT.

In this paper, the model $Qos(CS)$ based on the semantic matching degree and the composite service quality is designed. Then, the fitness function $Fitness(CS)$ and the dynamic service composition and selection algorithm $IC\&S\_SWTS$ are designed. Finally, by comparing experiments with existing related algorithms, the validity of $IC\&S\_SWTS$ is proved. The algorithm $IC\&S\_SWTS$ takes into account the "global semantic matching" and "local semantic matching", and the assessment factor $Qos_B(CS)$ proposed here has a greater impact on the precision. Therefore, the new algorithm $IC\&S\_SWTS$ can have better performance in terms of time performance and precision.

References

[1] C. C. Aggarwal, N. Ashish, A. Sheth, "The Internet of Things: A Survey from the Data-centric Perspective," *Managing and Mining Sensor Data*, Springer US, 2013, pp. 383-428.

[2] M. Lan, J. Xu, Gao W, "Ontology Similarity Computation and Ontology Mapping Using Distance Matrix Learning Approach," *IAENG International Journal of Computer Science*, vol. 45, no. 1, pp. 164-176, 2018.

[3] X. Y. Chen, G. Y. Li, "Semantic Collaboration and Sensing as A Service in Semantic Web of Things," *International Journal on Smart Sensing and Intelligent Systems*, vol. 9, no. 2, pp. 997-1028, 2016.

[4] K. Huang, Y. Fan, W. Tan, "Recommendation in An Evolving Service Ecosystem based on Network Prediction," *IEEE Transactions on Automation Science and Engineering*, vol. 11, no. 3, pp. 906-920, 2014.

[5] S. G. Deng, J. W. Yin, Y. Li, et al, "A Method of Semantic Web Service Discovery based on Bipartite Graph Matching," *Chinese Journal of Computers*, vol. 31, no. 8, pp. 1364-1375, 2008.

[6] S. Y. Lin, C. H. Lai, C. H. Wu, et al, "A Trustworthy QoS-based Collaborative Filtering Approach for Web Service Discovery ," *Journal of Systems and Software*, vol. 93, pp. 217-228, 2014.

[7] H. Wang, B. Zhu, G. Li, et al, "A Fusion of Grandulation and Artificial Neural Network: A New Service Selection Method," *2016 9th International Symposium on Computational Intelligence and Design (ISCID) IEEE 2016*, pp. 342-347.

[8] Y. Y. Tang, W. Liu, F. J. Liu, et al, "Web Service Dynamic Composition based on Event Ontology," *Journal of Huazhong University of Science and Technology (Nature Science Edition)*, vol. 41, no. Z2, pp. 22-25, 2013.

[9] S. P. Liu, D. Y. Liu, H. Qi, et al, "Composing Semantic Web Service with Description Logic Rules," *Journal of Computer Research and Development*, vol. 48, no. 05, pp. 831-840, 2011.

[10] L. Chang, J. Liu, T. L. Gu, et al, "Semantic Web Service Composition based on Dynamic Description Logics," *Chinese Journal of Computers*, vol. 36, no. 12, pp. 2468-2478, 2013.

[11] Y. S. Liu , D. Zhu . Semantic Web Service Discovery based on Clustering and Bipartite Graph Matching," *Computer Engineering*, vol. 42, no. 2, pp. 157-163, 2016.

[12] D. Kelaidonis, A. Rouskas, V. Stavroulaki, et al, "A Federated Edge Cloud-IoT Architecture," *European Conference on Networks and Communications IEEE 2016*, pp. 230-234.

[13] N. Kumar, N. Chilamkurti, J. H. Lee, "Distributed Context Aware Collaborative Filtering Approach for P2P Service Selection and Recovery in Wireless Mesh Networks," *Peer-to-Peer Networking and Applications*, vol. 5, no. 4, pp. 350-362, 2012.

[14] P. Rodriguez-Mier, C. Pedrinaci, M. Lama, et al, "An Integrated Semantic Web Service Discovery and Composition Framework," *IEEE Transactions on Services Computing*, vol. 9, no. 4, pp. 537-550, 2015.

[15] S. Bansal, A. Bansal, G. Gupta, et al, "Generalized Semantic Web Service Composition," *Service Oriented Computing and Applications*, vol. 10, no. 2, pp. 111-133, 2016.

[16] J. Puttonen, A. Lobov, M. A. C. Soto, et al, "Planning-based Semantic Web Service Composition in Factory Automation," *Advanced Engineering Informatics*, vol. 29, no. 4, pp. 1041-1054, 2015.

[17] A. Bekkouche, S. M. Benslimane, M. Huchard, et al, "QoS-aware Optimal and Automated Semantic Web Service Composition with Users Constraints," *Service Oriented Computing and Applications*, vol. 11, no. 2, pp. 183-201, 2017.

[18] S. Chattopadhyay , A. Banerjee, "A Methodology for Search Space Reduction in QoS Aware Semantic Web Service Composition," *arXiv preprint arXiv:1809.07045*, 2019.

[19] D. Paulraj, S. Swamynathan, D.Chandran, et al, "Service Composition and Execution Plan Generation of Composite Semantic Web Services Using Abductive Event Calculus," *Computational Intelligence*, vol. 32, no. 4, pp. 711-737, 2016.

[20] M. E. Khanouche, F. Attal, Y. Amirat, et al, "Clustering-based and QoS-aware Services Composition Algorithm for Ambient Intelligence," *Information Sciences*, vol. 482, pp. 419-439, 2019.

[21] P. Y. Zhang, B. Huang, Y. M. Sun, "A Web Service Matching Mechanism based on Semantics and Qos-aspect," *Journal of Computer Research and Development*, vol. 47, no. 5, pp. 780-787, 2010.

[22] S. L. Liu, Y. X. Liu, F. Zhang et al, "A Dynamic Web Services Selection Algorithm with QoS Global Optimal in Web Services Composition," *Journal of Software*, vol. 18, no. 3, pp. 646-656, 2007.

[23] 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*, vol. 94, pp. 185-198, 2019.

[24] L. Bao, F. Zhao, M. Shen, et al, "An Orthogonal Genetic Algorithm for QoS-aware Service Composition," *The Computer Journal*, vol. 59, no. 12, pp. 1857-1871, 2016.

[25] 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*, vol. 73, no. 4, pp. 1387-1415, 2017.

[26] S. Y. Toh, M. Tang, "Software-as-a-Service Composition in Cloud Computing Using Genetic Algorithm," *International Conference on Neural Information Processing, Springer, Cham 2018*, pp. 542-551.

[27] A. S. da Silva, H. Ma, M. Zhang, "Genetic Programming for QoS-Aware Web Service Composition and Selection," *Soft Computing*, vol. 20, no. 10, pp. 3851-3867, 2016.

[28] C. Jatoth, G. R. Gangadharan, U. Fiore, et al, "QoS-aware Big Service Composition Using MapReduce based Evolutionary Algorithm with Guided Mutation," *Future Generation Computer Systems*, vol. 86, pp. 1008-1018, 2018.

[29] Y. Li, J. Hu, Z. Wu, et al, "Research on QoS Service Composition based on Coevolutionary Genetic Algorithm," *Soft Computing*, vol. 22, no. 23, pp. 7865-7874, 2018.

[30] H. Ma, A. Wang, M. Zhang, "A Hybrid Approach Using Genetic Programming and Greedy Search for QoS-aware Web Service Composition," *Transactions on Large-Scale Data-and Knowledge-Centered Systems XVIII. Springer, Berlin, Heidelberg 2015*, pp. 180-205.

[31] S. K. Gavvala, C. Jatoth, G. R. Gangadharan, et al, "QoS-aware Cloud Service Composition Using Eagle Strategy," *Future Generation Computer Systems*, vol. 90, pp. 273-290, 2019.

[32] A. Sch?tzle, M. Przyjaciel-Zablocki, S. Skilevic, et al, "S2RDF: RDF Querying with SPARQL on Spark," *Proceedings of the VLDB*, vol. 9, no. 10, pp. 804-815, 2016.

[33] H. Y. Wang, Y. Y. Bai, "Composite Service Selection Method Supporting Semantic Matching based on Bipartite Graph," *Journal of Southeast University(Natural Science Edition)*, vol. 44, no. 3, pp. 510-516, 2014.

[34] X. Y. Chen, G. Y. Li, H. Chen, et al, "The Method for Semantic Similarity based on Concept Distance," *Advances in Computer Science Research*, no. 91, pp. 243-248, 2019.

[35] K. Su, L. Ma, X. Guo, et al, "An Efficient Parameter-adaptive Genetic Algorithm for Service Selection with End-to-end QoS Constraints," *Journal of Computational Information Systems*, vol. 10, no. 2, pp. 581-588, 2014.

[36] Y. Dai, L. Yang, "Qos for Composite Web Service and Optimizing," *Chinese Journal of Computers*, vol. 29, no. 7, pp. 1167-1178, 2006.

[37] F. Lecue, N. Mehandjiev, "Seeking Quality of Web Service Composition in A Semantic Dimension," *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 6, pp. 942-959, 2011

[38] V. Ho-Huu, S. Hartjes, H. G. Visser, et al, "An Improved MOEA/D Algorithm for Bi-objective Optimization Problems with Complex Pareto Fronts and Its Application to Structural Optimization," *Expert Systems with Applications*, vol. 92, pp. 430-446, 2018.

[39] J. Wu, L. Chen, S. G. Deng, et al, "Qos-skyline based on Dynamic Service Selection," *Chinese Journal of Computers*, vol. 33, no. 11, pp. 2136-2146, 2010.

[40] Y. Y. Fanjiang, S. Yang, "Semantic-based Automatic Service Composition with Functional and Non-functional Requirements in Design Time: A Genetic Algorithm Approach," *Information and Software Technology*, vol. 56, no. 3, pp. 352-373, 2014.

[41] O. Kramer, *Genetic Algorithm Essentials, Springer*, 2017.

[42] P. K. N. Banu, S. Andrews, "Evaluation of Fitness Functions for Swarm Clustering Applied to Gene Expression Data," *Computational Intelligence in Data Mining-Volume 3. Springer, New Delhi*, 2015, pp. 571-581.

[43] Q. Wu, F. Ishikawa, Q. Zhu, et al, "QoS-aware Multigranularity Service Composition: Modeling and Optimization," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, no. 11, pp. 1565-1577, 2016.

[44] C. Rahmad, R. Ariyanto, D. R. Yunianto, "Brain Signal Classification Using Genetic Algorithm for Right-left Motion Pattern," *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 11, pp. 247-251, 2018.

[45] C. Wang, H. Ma, G. Chen, et al, "Evolutionary Multitasking for Semantic Web Service Composition," *arXiv preprint arXiv:1902.06370*, 2019.

[46] Z. Wang, J. Wang, B. Li, et al, "Online Cloud Provider Selection for QoS-sensitive Users: Learning with Competition," *IAENG International Journal of Computer Science*, vol. 43, no. 3, pp. 310-317, 2016.

[47] S. Li, L. Jiao, Y. Zhang, et al, "A Scheme of Resource Allocation for Heterogeneous Services in Peer-to-peer Networks Using Particle Swarm Optimization," *IAENG International Journal of Computer Science*, vol. 44, no. 4, pp. 482-488, 2017.