Comprehensive Quality-Aware Automated Semantic Web Service Composition

Chen Wang, Hui Ma, Aaron Chen
School of Engineering and Computer Science, Victoria University of Wellington, New Zealand
Email: chen.wang@ecs.vuw.ac.nz, hui.ma@ecs.vuw.ac.nz, aaron.chen@ecs.vuw.ac.nz

***Abstract*—Semantic web service composition has been a prevailing research area in recent years. There are two major governance challenges faced by researchers (semantic matchmaking and QoS optimisation). Semantic matchmaking aims to discover interoperable web services that can interact with each other to provide rich functionalities through language understanding and reasoning. QoS optimisation aims to optimise the non-functional requirement of service users. E.g., minimum cost, maximal reliability. Many scholars have looked into nonfunctional optimisation problems in QoS-aware web service composition applying AI planning and Evolutionary Computation techniques. To meet users' requirement, one often needs to consider both semantic matchmaking quality and QoS simultaneously. Existing works on web service composition often concerns either semantic service composition or QoS-aware service composition. However, there are few works concerns both quality of semantic matchmaking and quality of services at the same time and points out the benefits clearly. Therefore, In this paper, we consider two quality dimensions in a proposed more applicable comprehensive quality model for semantic web service composition and present an automated semantic web service composition approach that can generate service composition solutions that meet both functional and non-functional requirement. The results show our method can find better semantic matchmaking quality with a very slight trade off in QoS. Also, our method performs better in finding optimised solution with a trade off in execution time comparing an existing GP-based method.**

I. Introduction

Web service composition pertains to a combination of multiple web services to provide a value-added composite service that accommodates customers' arbitrarily complex requirements. This application is developed by integrating interoperable and collaborative functionalities over heterogeneous systems. Due to the increase of the number of large-scale enterprise applications, the number of Web services has increased substantially and unprecedentedly. Therefore, manual and semi-automated web service composition are considered to be less efficient while automated web service composition enjoys less human intervention, less time consumption, and high productivity.

Two most notable challenges for web service composition are (1) ensuring interoperability of services and (2) achieving QoS optimisation [1]. *Interoperability* of web services presents challenge (1) in syntactic and semantics dimensions. The syntactic dimension is covered by the XML-based technologies (such as $WSDL$, $SOAP$). The semantics aspect, on the other hand, demands further research. Through ontology-based semantics [2], web services can understand and better collaborate with each other. There are many ontologies languages and formats for semantic service descriptions, such as OWL-S, WSML, and SAWSDL [3], which make "machine understanding" possible through identifying and matching semantic similarity in input/output parameters of web services in heterogeneous environments. The second challenge (2) is related to finding *optimised solutions* to Quality of Service (QoS). This problem gives birth to *QoS-aware service composition* that considers the composition of service-level agreements (SLA) [4] involving a collection of SLA rules and policies for supporting QoS-based composition.

Existing works on service composition focus mainly on addressing the one of challenges above. A lot of works have been conducted to optimise the quality of compositions under a pre-defined abstract workflow, which is generally considered as *semi-automated Web service composition* approach. Meanwhile, many research works consider the possibility of generating a composite plan automatically in discovering and selecting suitable web services, which are considered to be NP-hard [5]. *Semantic web services composition* is distinguished from the syntactic service composition, with the hope of eliminating conflicts by the semantic level of web services. In the past few years, substantial works have been done on semantic web service composition [6], [7], [8]. However, few works have enabled truely automatic semantic web service composition, where both QoS and quality of semantic match making will be optimised simultaneously. To a given service request, the quality of service composition solution depends on both QoS and quality of semantic matchmaking. Therefore, a comprehensive quality model is needed to be addressed, and more methods are needed to be applied specifically on the problem that optimises semantic matchmaking quality and QoS simultaneously.

The overall goal of this paper is to *develop an comprehensive quality-aware automated semantic web service composition that satisfactorily optimises both functional and non-functional requirements*. Particularly, this paper extends existing works of QoS-aware service composition by considering both QoS optimisation and semantic matching optimisation in our proposed comprehensive quality model. Particle Swarm Optimisation (PSO) has show its promise in searching near-optimised service composition solution [9], we will propose a PSO-based service composition approach using our comprehensive quality model, which is also considered to be more applicable way to measure semantic matchmaking in automated semantic web service composition. We will achieve three objectives in this work as follows:

1) To propose a comprehensive quality model that address QoS and semantic matchmaking quality in considering different matching types with corresponding concept similarity. This method is consider to be a more applicable and effective way of finding an optimised solution in semantic web service composition.
2) To propose a PSO-based service composition algorithm that utilises the proposed quality model. To do that we will first propose a representation of semantic service composition, which can model the quality of semantic matchmaking and QoS together. A particle is then used to represent a queue of services that can be used to decoded into a near-optimised semantic service composition solution.
3) To evaluate the performance of comprehensive quality awareness semantic web service composition by utilising benchmark datasets from Web Services Challenge 2009 (WSC09) [10]. Also, we make a comparison between PSO-based and an existing GP-based approach employing our proposed comprehensive quality model.

II. Related Work

Semantic web service composition. Substantial work [9], [11], [12], [13], [14] on web service composition focused primarily on non-functional requirements consistently neglecting functional requirements. However few researchers addressed both functional and nonfunctional requirements at the same time in web service composition. To the best of our knowledge, [6], [7], [8] reported some recent attempts on service composition that considers both nonfunctional (QoS) and functional (semantic) aspects. Semantic matchmaking utilises Description Logic(DL) [15] to search for services that ensure semantic matching utilising an ontology of the corresponding domain.

The combination of enhancing planning graph (EPG) and clonal selection algorithm is introduced in the work [8] as an immune-inspired web service composition approach. During the clonal selection process, the antigen is represented as a fitness function, and the antibody is represented as a binary alphabet to encode EPG. In this work, semantic matchmaking is considered by measuring the quality of the similarity measure using information retrieval technique.

In [6], the quality of matchmaking problem is transferred to measure the quality of semantic links, one possible measure is applied to the degree of similarity using Common Description rate of a semantic link, where Extra Description and Least common subsume are required to be pre-calculated. Therefore, the quality of the semantic link is estimated as quality of matching types associated with their corresponding quality of Common Description rate. However, the weakness of semantic link quality is that calculation requires well and completely defined ontology in class, class axioms and properties. This makes it difficult to measure semantic matchmaking quality as it takes huge cost and time for the domain experts to establish required ontology. In comparison, we introduce a more applicable comprehensive quality model in a fully automated approach.

The work [7] concerns both functional and non-functional requirements in design time of semantic-based automatic service composition, where a GA-based approach with four unusual independent fitness functions are designed to solve the problems with these concerns. A sequence of fitness functions are used in the binary selection of chromosomes, rather than a single objective consisting of both functional and non-functional quality. Apart from that, semantic matching types are ignored in their quality evaluation of semantic matching.

QoS-aware web service composition. Evolutionary Computation techniques are widely used to solve optimisation problems, where the search space is big, so that near-optimised solutions can be found in reasonable time.

Gupta et al. [16] proposed an improved Genetic Algorithm for QoS-ware web service composition, where some population associated with lower fitness are discarded with bringing in new offerings in each generation. In [14], the author optimises the overall quality of service composition by using a fitness function, which is also liable for the correctness in functionality through penalising infeasible solutions. Also, this paper increase the mutation rate while encountering low diversity in the population and adopt a higher crossover probability while trapped in local optimisation. A hyper approach employs both a greedy search algorithm and genetic programming is introduced in [17] to generate locally optimised solution with functionality correctness. In particularly, the greedy search is used to generate directed acrylic graph as composite solutions, which is further transferred to tree structure using unfolding techniques for initialisation and mutation during the evolutionary process. A promising GraphEvol is proposed in [13], where web service composition are in a form of Directed Acyclic Graph(DAG) employing Graph-based evolutionary operators like crossover and mutation. To simplify the checking of constraints for solutions, an indirect PSO-based approach was introduced in [9]. A general graph is used as representation as composite solution in their PSO algorithm considering QoS optimisation.

Other non-EC techniques are also introduced in QoS-ware web service composition. A heuristic service composition method is proposed by Tang et al. [18] proposed, named local optimisation and enumeration, which is to filter a small number of promising candidates related to each task by local selection, and all the composite solutions are enumerated to reach the near-to-optimal.

However, composite solutions from [18], [9], [17], [13], [14] does not consider distinguished matchmaking quality in their fitness function, which might lead to over-general outputs to be produced by selected services, The finding is evidently supported by comparing two solutions employing different evaluation model in Sect. VI-A. In fact, customers' perspectives, application domains and ontology granularity all could have significant impact on the outputs requested by users. In some scenarios, the output is too broad to bear any specific meaning for the customers, even though those web services selected leads to a very good overall QoS. Therefore, we fill the gap by considering different matching types and parameter-related concept similarity to determine the overall quality of web service composition. Also, Weighed graphs are utilised as representationd of semantic web service composition decoded from optimised service queue using PSO adopted from [9].

III. Problem Description

The purpose of web service composition is to accomplish an arbitrarily complex task fulfilling customer's requirement, which could be denoted as a composite goal: $Comp.G(F(T_{Input}, T_{Output}), NF(T_{QoS}))$. This overall composite goal is demonstrated in two parts. The first functional part focus on transfering a given task input or input set to the desired Task output or output set. It typically refers to users' functional requirement. Another non-functional part specifies the acceptable level of composite quality of service. To accomplish the composite goal, two stages are involved: services discovery and service selection. Firstly, service discovery is to find matched web service: $S_n(F(S_{Input}, S_{Output}), NF(S_{QoS}))$ from a Service Repository: $S = \{S_1, S_2, ..., S_n\}$ with the given T_{Input}. If no atomic web service could satisfy the composite goal, a combination of web services will be found to meet $Comp.G$. To ensure the composite solution returns the desired $Comp.G$, we consider a comprehensive quality model for service discovery and selection, where challenges (1) and (2) mentioned in Sect. I are also addressed in Sect. IV.

A. Semantic Web Service matchmaking Type

The semantic service matchmaking aims to discover appropriate services from service repository in view of customers' functional requests. A semantic web service is defined by $S(F(S_{Input} \in C_1, S_{Output} \in C_2), NF(S_{QoS}))$, where both Input and Output are linked to concept C_1 and C_2 in an ontology (O) respectively, satisfying $O = \{C, Taxonomy\}$. A web service matching process is to match the output and input concepts of two services according to the Taxonomy within an Ontology (O). To measure the quality of semantic matchmaking, different matching levels are typically considered in the literature [19]. To understand these levels, let us define two web services associated with concept-related parameters in a particular domain. S_1 $(F(S_{Input} \in C_1, S_{Outputs} \in C_2), NF(S_{QoS}))$ and S_2 $(F(S_{Input} \in C_3, S_{Output} \in C_4), NF(S_{QoS}))$ and an Ontology(O) with C_1, C_2, C_3, and C_4. The matching levels to be considered are:

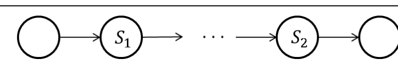
- *Exact* ($\equiv$): Output of Web service S_1 and Input of Web service S_2 are Exact match ($S_{output} \in S_1 \equiv S_{input} S_2$), if Concept C_2 and Concept C_3 are equivalent.
- *Plugin* ($\sqsubseteq_n$): Output of Web service S_1 and Input of Web service S_2 are Plugin match ($S_{output} \in S_1 \sqsubseteq_n S_{input} \in S_2$), if Concept C_2 is a sub-concept of Concept C_3, and $n = \{1, 2, ..., n\}$ presents the levels of children concepts ($n = 1$ stands for direct children).
- *Subsume* ($\sqsupseteq_n$): Output of Web service S_1 and Input of Web service S_2 are Subsume matched($S_{output} \in S_1 \sqsupseteq_n S_{input} \in S_2$), if Concept C_2 is a sub-concept of Concept C_3, and $n = \{1, 2, ..., n\}$ presents the levels of parent concepts ($n = 1$ stands for direct parent).
- *Fail* ($\perp$). Output of Web service S_1 and Input of Web service S_2 are not matched (Fail) ($S_{output} \in S_1 \perp S_{input} \in S_2$), if none of the previous matches discovered.

B. Quality of Service and Composition Constructs

Currently, most of the optimisation problems [20], [21], [17], [22] in web service composition are focusing on QoS, which covers aspects in non-functional requirements. This problem has been explored in both single objective and multi-objectives optimisation problems. Customers prefer lowest execution cost with highest response time and reliability that could be optimised simultaneously. According to [23], four most often considered QoS parameters are as follows:

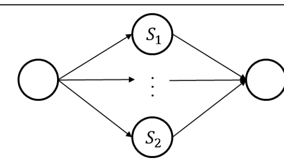
- *Response time* (T) measures the expected delay in seconds between the moment when a request is sent and the moment when the results are received.
- *Cost* (C) is the amount of money that a service requester has to pay for executing the web service
- *Reliability* (R) is the probability that a request is correctly responded within the maximum expected time frame.
- *Availability* (A) is the probability that a web service is accessible.

The aggregation value of QoS attributes for web services composition varies with respect to different constructs, which reflects how services associated with each other in a service composition [23]. Here we consider two composite constructs: sequence and parallel constructs, in building the composite web service. The QoS calculation models are described as follows:



$$T = \sum_{n=1}^{m} t_n \quad C = \sum_{n=1}^{m} c_n \quad A = \prod_{n=1}^{m} a_n$$
$$R = \prod_{n=1}^{m} r_n$$

Fig. 1. Sequence construct and calculation of its QoS properties [14].



$$T = MAX\{t_n | n \in \{1, \ldots, m\}\}$$
$$C = \sum_{n=1}^{m} c_n \quad A = \prod_{n=1}^{m} a_n \quad R = \prod_{n=1}^{m} r_n$$

Fig. 2. Parallel construct and calculation of its QoS properties [14].

1) Sequence construct: The composite web service executes each atomic service associated with a sequence construct in a definite sequence order. The aggregation value for total time (T) and total cost (C) is as the sum of time and cost of web services involved respectively. The overall availability and reliability in a sequence construct are calculated by multiplying their corresponding availability and reliability of each web service in probability theory. This construct is shown in Fig. 1.

2) Parallel construct: Web services in a parallel construct are executed concurrently. The QoS aggregation value for total cost, availability and reliability are the same as these in sequence construct while the Total time (T) is determined by the most time-consuming path in the composite flow of the solution. This construct is presented in Fig. 2.

IV. Comprehensive quality-aware Semantic Automated Web Service Composition

In this section, we propose a comprehensive quality model for automated semantic service composition, and optimise both semantic matchmaking quality and QoS. PSO has shown its efficiency in solving combinatorial optimisation problems [24]. Therefore, we will employ a PSO-based approach, which is considered to be simple and efficient without penalising or repairing that often required by GP [9]. Fig. 3 shows the overview of our approach with five steps. Step 1: The composite process is triggered by a composite goal defined in Subsection III, which describes customers requirements both functional and non-functional. Step 2: This composition goal are used to discover all relevant web services, which lead to a shrunken service repository that is subsequently used by PSO as a searching space. Step 3: A weighted graph representation is randomly built up from the an initial service queue that mapped to the particle's location, interleaving with semantic matchmaking process utilising ontology-based index cache. In the weighted graph, graph edges are assigned with semantic matchmaking quality as weights. Step 4: The fitness value of the weighted graph is evaluated to update the position of particle under PSO algorithm in Sect. IV-F, where the position is mapped to the index of service queue. later on, the updated service queue is used to decode a new weighted graph as the composition solution. Step 4. Lastly, the best position found in the searching space is selected and decode into the final optimised solution. This PSO-based approach is similar to [9], but we employ weighted graphs as a different solution presentation.

A. Semantic Matchmaking

To perform semantic matchmaking, we transfer a function match between $S_1 : S_{output} \in C_1$ and $S_2 : S_{output} \in S_b$ to a pair of concept match demonstrated in Sect. III-A. The matching process attempts to determine semantic matching between the source concepts of C_1 and the target concepts of C_2. Meanwhile, the quality of matched concepts are calculated in the quality model in subSection IV-C. The purpose of semantic matchmaking is to find more component services that could also potentially satisfy the quality of QoS with good functional quality.

The semantic matchmaking is achieved by utilising OWL2 and OWL-S or other semantic markup languages for web services. In this paper, we use MECE (Mediation Contract Extension) [25] and OWL-DL. MECE is considered to be an alternative semantic annotation for WSDL. MECE defines the service-related inputs and outputs with parameter-related concepts. OWL-DL is a sublanguage of OWL extended from RDF. It specifies semantic information of concepts involved in MECE.

B. Ontology-based Index Cached Optimisation

In PSO, particles represent weighted graphs with edges and vertices associated with quality of semantic matchmaking and QoS respectively, the bottlenecks of generating weighted graph lie in building edges and nodes, which are related to the cost of semantic quality calculation and the size of service repository respectively. To effectively construct weighted graphs, we pre-calculate semantic matchmaking quality. The key idea of the index is to create a mapping from the output-related concepts to potentially matched input-related concepts while considering different levels of match types. Meanwhile, this index size could also be reduced by only considering the concepts related in the shrunken service repository, which means the index cache is filtered by those task-relevant web services. This optimised cache also contributes to less and constant time for weighted graphs building through the whole evolutionary process.

C. Comprehensive Quality Model and Aggregation Matrix

In this paper, we propose a comprehensive quality model to evaluate the overall quality of semantic web service composition. This model overcome the disadvantages of current prevailing QoS-aware optimisation that ignores quality of semantic matchmaking.

Semantic matchmaking model. Due to the discretisational characteristics of different match types and values assigned to matching types that driven by the cost of data integration and manipulation [6], partial ordering match types are considered to be one factor for the semantic matchmaking quality. For example, Exact matching type demands less time for computation compared to that of Plugin match. Another factor in our proposed model is concept similarity, which could be evaluated based on the edge counting method defined in [26]. This formula (1) is used to estimate the similarity between concept-related parameters for selecting web services. Therefore, given the quality match type and the concept similarity of two parameters-related concepts, the semantic matchmaking quality of matched parameters is defined by Formula (2), where the value of $q(p_{mt})$ follows the same settings in [6]. Particularly, 1 (Exact), 0.75 (Plugin), 0.5 (Subsume) or 0.25 (Intersection).

$$q(p_s)=\frac{2N \cdot e^{-\lambda L/D}}{N_1+N_2} \tag{1}$$

$$q(p_{sm}) \doteq (q(p_{mt}),\ q(p_s)) \tag{2}$$

Further more, edges represent services connections in our weighted graph representations, where assigned weight value $q(e_{sm})$ is considered to be semantic matching quality on edge level according to parameter aggregations. The weight value $q(e_{sm})$ is defined in 3, where $q(e_{mt})$ and $q(e_s)$ are the average value of concept-related parameters quality in $q(p_{mt})$ and $q(p_s)$ respectively.

$$q(e_{sm}) \doteq (q(e_{mt}),\ q(e_s)) \tag{3}$$

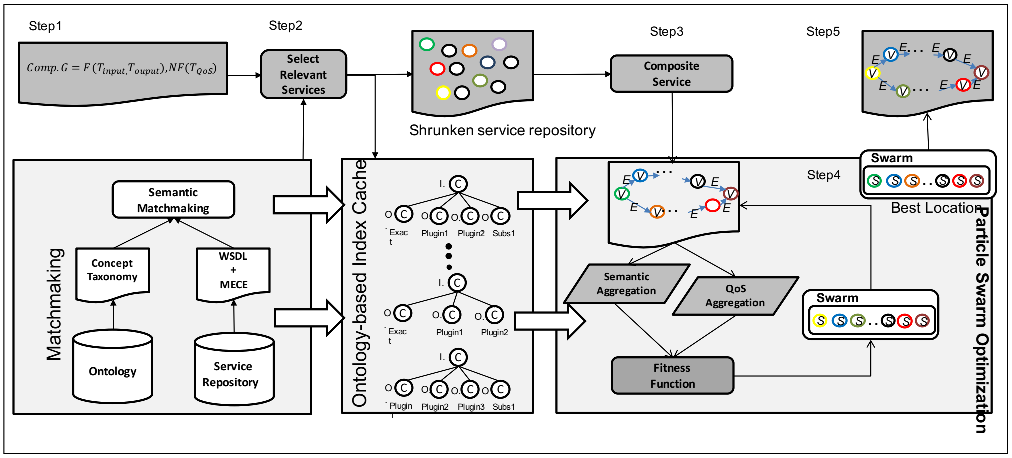


Fig. 3. Overview of POS-based automated semantic web service composition approach.

TABLE I
QUALITY AGGREGATE MATRIX FOR SEMANTIC WEB SERVICE COMPOSITION

Composition Construct			Sequence	Parallel
Quality	F	$Q(e_{mt})$	$\prod_{n=1}^{m} q(e_{mt})$	$\prod_{n=1}^{m} q(e_{mt})$
		$Q(e_s)$	$(\sum_{n=1}^{m} q(e_s))/m$	$(\sum_{n=1}^{m} q(e_s))/m$
	NF	$Q(v_a)$	$\prod_{n=1}^{m} q(v_a)$	$\prod_{n=1}^{m} q(v_a)$
		$Q(v_r)$	$\prod_{n=1}^{m} q(v_r)$	$\prod_{n=1}^{m} q(v_r)$
		$Q(v_c)$	$\sum_{n=1}^{m} q(v_c)$	$\sum_{n=1}^{m} q(v_c)$
		$Q(v_t)$	$\sum_{n=1}^{m} q(v_t)$	$max(q(v_t))$

Comprehensive quality model. Compared to QoS evaluation model, the comprehensive quality model is established to investigate both functional and non-functional requirements. The comprehensive quality of our service composition representation refers to QoS of service vertices and Semantic matching quality of Edges in weighted graphs. Consequently, the comprehensive quality model is defined in Formula (4), which could be further broken down into Formula (5).

$$q_{cq} \doteq (q(e_{sm}),\ q(v_{QoS})) \tag{4}$$

$$q_{cq} \doteq (q(e_{mt}),\ q(e_s),\ q(v_a),\ q(v_r),\ q(v_c),\ q(v_t)) \tag{5}$$

Quality aggregate matrix. The quality aggregation is defined based on the constructs of composite web services, in consideration of functional and non-functional properties. The quality on construct level is further calculated by following the rules summarised in Table I.

D. Composition Weighted Graph

We defined our semantic web service composition solution as a weighted graph $WG = (V, E)$, where V is a set of services as vertex: $V = [S1, S2...Sn]$ and E is a set of edges $E = e_1, e_2, ...e_n$. Each e is associated with $q(e_{sm})$ as weight value that mapped to a pair of quality values $q(e_{mt})$ and $q(e_s)$ on the edge level, and e_m is expressed as $(S_a, S_b) = q_{mt}, q_s$. Here we provide an example of the web service composition, which is described in Fig. 4. The data of composite web service flows from the start to the end, where five web services involved and linked with each other using edge connections. Besides that, $q(s_{mt})$ and $q(s_s)$ are calculated for all edges $e_1, e_2, ...e_n$ assigned as weights.

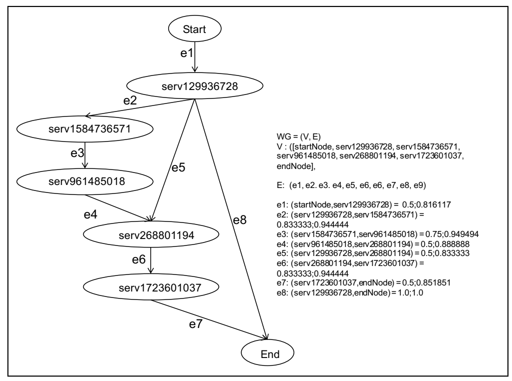


Fig. 4. Composition weighted graph solution

E. Fitness Calculation

In real life, given a unique and optimised solution is always easier for customers to pick up directly when many quality criteria involved into decision making, rather than provided a set of solutions. Therefore, it is very practical to define a single fitness as a weighted sum of all the quality related components in Formula (6), where weights setting is very flexible, it could be adjusted according to users' preferences. The fitness value of 1 means the best comprehensive quality and 0 means the worst. For this purpose, MT, S, A, R, T, and C must be normalised so that the fitness value falls within the range from 0 to 1. Therefore, the service composition problem in this paper is treated as a fitness maximisation problem.

$$Fitness = w_1\hat{MT} + w_2\hat{S} + w_3\hat{A} + w_4\hat{R} + w_5(1-\hat{T}) + w_6(1-\hat{C}) \tag{6}$$

where $\sum_{i=1}^{6} w_i = 1$

$$\hat{Q}_k = \begin{cases} \frac{Q_k - Q_{k,min}}{Q_{k,max} - Q_{k,min}} & \text{if } Q_{k,max} - Q_{k,min} \neq 0. \\ 1 & \text{otherwise.} \end{cases} \tag{7}$$

where $k = 1, 2, 3,$ and 4, where Q_1 as MT, Q_2 as S, Q_3 as A, and Q_4 as R.

$$\hat{Q}_j = \begin{cases} \frac{Q_{j,max} - Q_j}{Q_{j,max} - Q_{j,min}} & \text{if } Q_{j,max} - Q_{j,min} \neq 0. \\ 1 & \text{otherwise.} \end{cases} \tag{8}$$

where $j = 1$, and 2, where Q_1 as T and Q_2 as C.

F. QoS-aware Semantic Web Service Composition Algorithm

ALGORITHM 1. Steps of the PSO-based Web service composition technique.

Input : relevant Web services rws
Output: Optimised service queue $queue$,Optimised Weighted Graph $OPTWG$

Map each relevant service to an index in the particle's position vector;
Randomly initialise each particle in the swarm;
while *max. iterations not met* **do**
  foreach *particles in the swarm* **do**
    $queue \leftarrow$ particle's position vector;
    $WG \leftarrow generateWeightedGraph()$;
    Calculate the WG fitness value;
    if *fitness value better than pBest* **then**
      Assign current fitness as new pBest;
    else
      Keep previous pBest;
  Assign best particle's *pBest* value to *gBest*, if better than gBest;
  Calculate the velocity of each particle;
  Update the position of each particle;
return $OPTWG$;

ALGORITHM 2. Create a composition weighted graph from a queue.

Procedure `generateWeightedGraph()`
  Input : Task inputs I, task outputs O, Optimised service queue $queue$, IndexCache $IndexCache$
  Output: Weighted Graph WG
  $WG \leftarrow null$;
  $WG \leftarrow$ new $endNode()$, new $startNode()$;
  $OutputSet \leftarrow \{I\}$;
  while *all* $O \notin OutputSet$ *and* $queue! = null$ **do**
    foreach ws *in* $queue$ **do**
      if $ws.inputs \in OutputSet$ **then**
        foreach I *in* $ws.inputs$ **do**
          $p_{mt} \leftarrow$ query $IndexCache$;
          $p_s \leftarrow$ query $IndexCache$;
        $e_{sm} \leftarrow$ aggregation(p_{mt},p_s);
        $WG.edge \leftarrow e_{sm}$;
        $WG \leftarrow$ new $wsNode()$;
        $OutputSet$ add $\{ws.outputs\}$;
        $queue$.remove ws;
  remove $danglingnodes$;
  remove $danglingedges$;
  return WG;

The overall algorithm investigated here is made up of the PSO-based web service composition algorithm 1 and the decoding algorithm 2. In algorithm 1, the idea is to translate the particle location produced by PSO into a service queue as an indirect representation, such that finding the best fitness of the weighted graph is to discover the optimised location of the particle in the search space. In PSO, the dimension of each particle equals to the number of relevant web services. The index of each services is mapped to a separate location component in a particle. Services in a queue follow the ascending order, from which we decode a weighted graph using Algorithm 2. It is a simple forward graph building algorithm, and this method can lead to more services and edges connected to the graph as redundancies that must be removed. Also, semantic quality value are assigned to all the edges, which is calculated from a quality aggregation function from all involved concept-related parameters.

V. Experiment Design

In this section, a quantitative evaluation approach is adopted in our experiment design. The objectives of the evaluation are to (1) measure the effectiveness of the comprehensive quality model in automated semantic web service composition approach; (2) explore the impacts of the semantic matchmaking that contributes to overall composition quality; and (3) compare solutions generated by QoS-ware approach with our method; and (4) compare our method with one existing GP-based approach.

We utilise benchmark dataset web service challenge 2009 (WSC09) [10] to perform the evaluation. WSC09 provides problems with five tasks corresponding to variable number of services, and ontologies. Therefore, it is a challenge dataset for measuring the scalability of our quality evaluation model. Table II presents the features of the WSC09 dataset. The number of concepts, individuals in the ontology and services in each data set is shown in the second, third, fourth column respectively. Also, we extend all the datasets with QoS attributes from service providers to enable our evaluation.

TABLE II
FEATURES OF THE WSC09 DATASETS

Dataset	No.Concept	No.Individual	No.Service
WSC09 01	1578	3102	572
WSC09 02	12388	24815	4129
WSC09 03	18573	37316	8138
WSC09 04	18673	37324	8301
WSC09 05	31044	62132	15211

We run the experiment on computing grid comprising of 170 NetBSD (Unix operating system) workstations operated by the Sun Grid Engine. The parameters were chosen based on general settings from [27] for our PSO-based approach, In particular, PSO population size is 30 with 100 generations. We run 30 times independently for each dataset. We configure weight of fitness function to properly balance functional side and nonfunctional side. Therefore, w_1 and w_2 are set equally to 0.25, and w_3, w_4, w_5, w_6 are all set to 0.125 accordingly. However, the settings of these weight values does not impact the method. In general, weight settings are adjusted according to users' preferences.

TABLE III
MEAN QUALITY FOR COMPREHENSIVE QUALITY-AWARE METHODS AND QOS-AWARE APPROACH

WSC09		QoS-aware Evaluation	Comprehensive Quality Evaluation
Task1	Q_{mt}	0.189787 ± 0.039278686	0.221862 ± 0.009582 ↑
	Q_s	0.884962 ± 0.014140	0.894082 ± 0.009206 ↑
	Q_{QoS}	0.278730 ± 0.007786	0.280222 ± 0.008212
Task2	Q_{mt}	0.001795 ± 0.000719	0.001977 ± 0.001566 ↑
	Q_s	0.906971 ± 0.005855	0.923970 ± 0.006898 ↑
	Q_{QoS}	0.239979 ± 0.000578 ↑	0.238596 ± 0.001264
Task3	Q_{mt}	0.158526 ± 0.014028	0.245830 ± 0.007761 ↑
	Q_s	0.949109 ± 0.002331	0.972765 ± 0.002980 ↑
	Q_{QoS}	0.247002 ± 0.000661 ↑	0.245631 ± 0.000431
Task4	Q_{mt}	0.000000 ± 0.000000	0.000004 ± 0.000002 ↑
	Q_s	0.879514 ± 0.007456	0.920733 ± 0.000001 ↑
	Q_{QoS}	0.242297 ± 0.000507 ↑	0.236677 ± 0.002211
Task5	Q_{mt}	0.000042 ± 0.000030	0.000078 ± 0.000020 ↑
	Q_s	0.915933 ± 0.012888	0.927678 ± 0.002578 ↑
	Q_{QoS}	0.238189 ± 0.000240 ↑	0.237485 ± 0.000328

VI. RESULTS AND ANALYSIS

A. Comparison Test with QoS Evaluation Model

In this section, we analyse the composition solution generated by using our approach comparing with QoS-aware approach. First, we look at mean value of Q_{mt}, Q_s and Q_{QoS} at optimum at the 100th generation for two approaches, shown in Table III. The QoS-aware approach record Q_{mt}, Q_s and utilise fitness function $Fitness = w_1\hat{A} + w_2\hat{R} + w_3(1-\hat{T}) + w_4(1-\hat{C})$ when $\sum_{i=1}^{4} w_i = 1$, but MT and S are recorded, and $Q_(QoS)$ is normalised from 0 to 0.5 to make it comparable to Q_{QoS} in our approach. We observe an interesting pattern from Table III using statistic analysis: mean value of Q_{mt} and Q_s at optimum by our approach is consistently higher than those by QoS-aware approach. Meanwhile, Q_{QoS} generated by QoS-aware approach can obtain a slightly higher value than that of our approach. In conclusion, we can perceive that our evaluation could find out better functional quality with a reasonable trade off in QoS.

Second, to compare the results generated from two evaluation approaches, we demonstrate an example solution that shows the differences in composite web services obtained through two different methods. Fig. 5 (1) and (2) show two composition weighted graphs as the two optmised solutions to Task 3 with (1) QoS-aware approach and (2) Comprehensive quality-aware method respectively. Two approaches generate exactly the same service workflow structure where those service vertices and edges denoted in red are different. We make a comparison of the quality among these different edges (e_1 to $e4$) associated service vertices in terms of quality of semantic matchmaking and QoS attributes in Fig. 5 (3). We also look at ΔQ which reveals the amount of variation on quality between two methods, where the positive values means the benefits gained under our approach while the negative values means the trade-off. To demonstrate the benefits of the positive value, we analyse the smallest positive ΔQ corresponding to e_4 and demonstrate how the output and required input are different under two approaches in Fig. 5 (4). $Ser1640238160$ and $Ser1947554374$ are selected service vertices with output concept-related parameters $Inst795998200$ and $Inst582785907$ corresponding to two concepts $Con103314376$ and $Con2037585750$ respectively, and $Inst658772240$ are the required parameter related to concept $Con2113572083$. There exist $Inst795998200 \in Con103314376 \sqsubseteq_2 Inst658772240 \in Con2113572083$ and $Inst582785907 \in Con2037585750 \sqsubseteq_3 Inst658772240 \in Con2113572083$. It is obvious that our approach selects the service providing $Inst795998200$ that are closer to the users' requirements suggesting that our method can produce better semantic matchmaking quality.

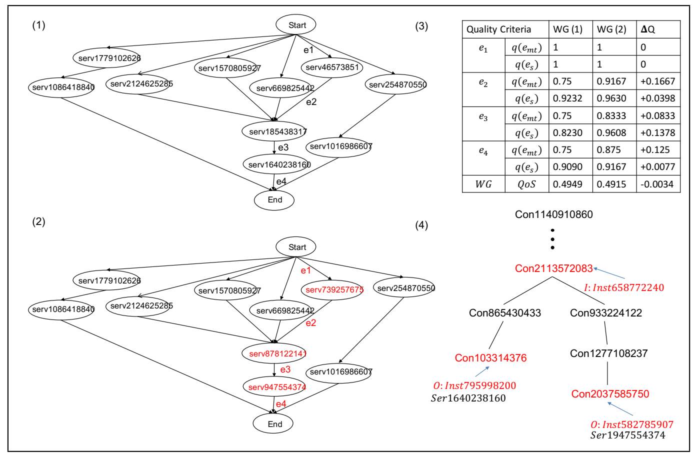


Fig. 5. Example Comparison of solutions to Task 3 under different approaches.

B. Convergence Test

To analyse the effectiveness of our approach, we study the convergence rate of the proposed method in this section to understand the convergence rate of five tasks in WSC09. We analysis the performances during the whole evolutionary process in Fig. 6, in which experiment results on five tasks are arranged in four groups consisting of average fitness, average matchType quality, average similarity quality and average QoS for generation 0-99 with optimum.

Firstly, the average fitness value with optimum is calculated by averaging the best fitness found in each generation over 30 independent runs. We can see that there is a significant increase in the fitness values towards optimum between generation 0 and generation 15-25, the remaining generation continues to produce a steady but moderate improvement in the fitness value and eventually reach a plateau with no further improvements can be observed. The same behaviour is observed over the rest tasks.

We also investigate the variation of quality of semantic matchmaking where average matchType quality with optimum and average similarity quality with optimum are studied in second and third column groups of Fig. 6. Similar to the fitness values, clear evidence of fast convergence can be observed with respect to theses quality values. This observation is also consistent for the improvement of mean QoS quality with optimum shown in the last column in Fig 6. Additionally, we don't see too much trade-off from the QoS as the a consistently increase in semantic matchmaking quality is observed.

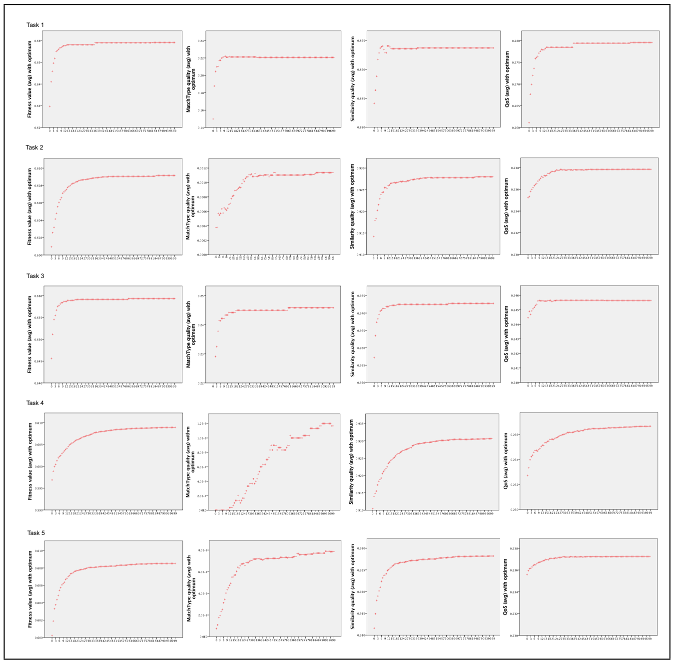


Fig. 6. Average Fitness, Average MatchType Quality, Average Similarity quality and Average QoS per generation with comprehensive quality optimum

C. Comparison Test with GP-based approach

We compare one GP-based approach [17] with our PSO method, where the individuals are tree structure with strict constrains transferred from a DAG employing unfolding techniques. We further mark all the service nodes in the tree representation with all their outgoing edges for considering semantic matchmaking quality, and those edges information are maintained correctly in the crossover and mutation. At last, we evaluate both semantic matchmaking quality and QoS simultaneously using the discussed the comprehensive quality model to investigate the performances of two approaches. To make a fair comparison with our PSO approach considering the same number of evaluations (3000 times), we adjust the parameters settings for GP approach [17] to 30 individual for 100 generations.

The Table IV shows the mean fitness values accomplished by two methods and Table V shows the average execution time required by them. We employ statistical analysis to test the significant differences in mean fitness value and execution time. The results show that the PSO-based approach performs better in four of five tasks. However, there is also a trade-off between optimised solution and execution time. We can see the PSO-based approach takes longer execution time, it due to that every individual in each generation must be decoded into a solution from optimised queue, and this process is very time-consuming. However, In GP, the initialisation first population is costly in time as transformation process from a DAG, but following new individuals are generated through mutation and crossover, which demands much less time comparing decoding process.

At last, We also compare the average fitness value with optimum through the evolutionary process for both two approaches over 30 independent runs. Fig. 7 is an example of convergence rate from Task 3, where the behaviour of PSO-based approach presents a very clear evidence of fast convergence and reaches a better-optimised solution, while the performance of GP is barely satisfactory, as GP-based method improves its fitness value gradually.

TABLE IV
MEAN FITNESS RESULTS FOR COMPARING GP-BASED APPROACH

Dataset	PSO-based approach	GP-based approach
WSC09 01	$0.559207 \pm 0.012780 \uparrow$	0.518411 ± 0.018470
WSC09 02	0.470083 ± 0.001106	$0.471594 \pm 0.002436 \uparrow$
WSC09 03	$0.559207 \pm 0.012780 \pm \uparrow$	0.552947 ± 0.007094
WSC09 04	$0.468942 \pm 0.001670 \pm \uparrow$	0.442934 ± 0.010154
WSC09 05	$0.469424 \pm 0.000800 \pm \uparrow$	0.256425 ± 0.038916

TABLE V
MEAN TIME RESULTS FOR COMPARING GP-BASED APPROACH

Dataset	PSO-based approach	GP-based approach
WSC09 01	12907 ± 8105	$5168 \pm 520 \downarrow$
WSC09 02	292346 ± 96727	$27483 \pm 3083 \downarrow$
WSC09 03	$12908 \pm 8105 \downarrow$	28053 ± 3202
WSC09 04	6553640 ± 1468204	$540418 \pm 186540 \downarrow$
WSC09 05	1208193 ± 266916	$180408 \pm 435247 \downarrow$

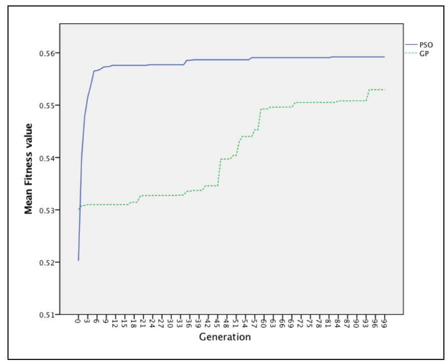


Fig. 7. An example of comparing convergency rate of PSO and GP approaches for task 3

VII. CONCLUSION

This work introduces a comprehensive evaluation model employed our PSO-based method for QoS-aware automated semantic automated web service composition that combines the quality of semantic matchmaking with QoS. The results show that our approach obtain better functional quality with a reasonable trade-off in QoS. Also, we compare one GP-approach with our PSO-based method to show our performance results in finding better-optimised solution. Future works could investigate other direct or indirect representations of semantic web service composition on semantic web service composition, and look into other EC techniques to evaluate their performance. Also, we could reshape the problem on providing customers with a set of solutions to choose from, rather than a single solution, so research question is redefined as multi-objectives or many objective problems.

REFERENCES

[1] D. Fensel, F. M. Facca, E. Simperl, and I. Toma, *Semantic web services*. Springer Science & Business Media, 2011.

[2] D. O'Leary, "Review: Ontologies: A silver bullet for knowledge management and electronic commerce," *The Computer Journal*, vol. 48, no. 4, pp. 498–498, 2005.

[3] C. J. Petrie, *Web Service Composition*. Springer, 2016.

[4] A. Sahai, V. Machiraju, M. Sayal, A. Van Moorsel, and F. Casati, "Automated sla monitoring for web services," in *International Workshop on Distributed Systems: Operations and Management*. Springer, 2002, pp. 28–41.

[5] M. Moghaddam and J. G. Davis, "Service selection in web service composition: A comparative review of existing approaches," in *Web Services Foundations*. Springer, 2014, pp. 321–346.

[6] F. Lécué, "Optimizing qos-aware semantic web service composition," in *International Semantic Web Conference*. Springer, 2009, pp. 375–391.

[7] Y.-Y. FanJiang and Y. Syu, "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.

[8] C. B. Pop, V. R. Chifu, I. Salomie, and M. Dinsoreanu, "Immune-inspired method for selecting the optimal solution in web service composition," in *International Workshop on Resource Discovery*. Springer, 2009, pp. 1–17.

[9] A. S. da Silva, Y. Mei, H. Ma, and M. Zhang, "Particle swarm optimisation with sequence-like indirect representation for web service composition," in *European Conference on Evolutionary Computation in Combinatorial Optimization*. Springer, 2016, pp. 202–218.

[10] S. Kona, A. Bansal, M. B. Blake, S. Bleul, and T. Weise, "Wsc-2009: a quality of service-oriented web services challenge," in *2009 IEEE Conference on Commerce and Enterprise Computing*. IEEE, 2009, pp. 487–490.

[11] S. Bansal, A. Bansal, G. Gupta, and M. B. Blake, "Generalized semantic web service composition," *Service Oriented Computing and Applications*, vol. 10, no. 2, pp. 111–133, 2016.

[12] P. R. Mier, C. Pedrinaci, M. Lama, and M. Mucientes, "An integrated semantic web service discovery and composition framework," 2015.

[13] A. da Silva, H. Ma, and M. Zhang, "Graphevol: A graph evolution technique for web service composition," in *Database and Expert Systems Applications*, ser. Lecture Notes in Computer Science, Q. Chen, A. Hameurlain, F. Toumani, R. Wagner, and H. Decker, Eds. Springer International Publishing, 2015, vol. 9262, pp. 134–142.

[14] Y. Yu, H. Ma, and M. Zhang, "An adaptive genetic programming approach to qos-aware web services composition," in *2013 IEEE Congress on Evolutionary Computation*. IEEE, 2013, pp. 1740–1747.

[15] F. Baader, *The description logic handbook: Theory, implementation and applications*. Cambridge university press, 2003.

[16] I. K. Gupta, J. Kumar, and P. Rai, "Optimization to quality-of-service-driven web service composition using modified genetic algorithm," in *Computer, Communication and Control (IC4), 2015 International Conference on*. IEEE, 2015, pp. 1–6.

[17] H. Ma, A. Wang, and M. Zhang, "A hybrid approach using genetic programming and greedy search for qos-aware web service composition," in *Transactions on Large-Scale Data-and Knowledge-Centered Systems XVIII*. Springer, 2015, pp. 180–205.

[18] L. Qi, Y. Tang, W. Dou, and J. Chen, "Combining local optimization and enumeration for qos-aware web service composition," in *Web Services (ICWS), 2010 IEEE International Conference on*. IEEE, 2010, pp. 34–41.

[19] M. Paolucci, T. Kawamura, T. R. Payne, and K. Sycara, "Semantic matching of web services capabilities," in *International Semantic Web Conference*. Springer, 2002, pp. 333–347.

[20] Y. Feng, L. D. Ngan, and R. Kanagasabai, "Dynamic service composition with service-dependent qos attributes," in *Web Services (ICWS), 2013 IEEE 20th International Conference on*. IEEE, 2013, pp. 10–17.

[21] Z. Huang, W. Jiang, S. Hu, and Z. Liu, "Effective pruning algorithm for qos-aware service composition," in *2009 IEEE Conference on Commerce and Enterprise Computing*. IEEE, 2009, pp. 519–522.

[22] A. S. da Silva, H. Ma, and M. Zhang, "A graph-based particle swarm optimisation approach to qos-aware web service composition and selection," in *2014 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2014, pp. 3127–3134.

[23] L. Zeng, B. Benatallah, M. Dumas, J. Kalagnanam, and Q. Z. Sheng, "Quality driven web services composition," in *Proceedings of the 12th international conference on World Wide Web*. ACM, 2003, pp. 411–421.

[24] Y. Fukuyama, "Fundamentals of particle swarm optimization techniques," *Modern heuristic optimization techniques: theory and applications to power systems*, pp. 71–87, 2008.

[25] S. Bleul, D. Comes, M. Kirchhoff, and M. Zapf, "Self-integration of web services in bpel processes," in *Proceedings of the Workshop Selbstorganisierence, Adaptive, Kontextsensitive verteilte Systeme (SAKS)*, 2008.

[26] K. Shet, U. D. Acharya *et al.*, "A new similarity measure for taxonomy based on edge counting," *arXiv preprint arXiv:1211.4709*, 2012.

[27] Y. Shi *et al.*, "Particle swarm optimization: developments, applications and resources," in *evolutionary computation, 2001. Proceedings of the 2001 Congress on*, vol. 1. IEEE, 2001, pp. 81–86.