An Optimal Approach to the QoS-Based WSMO Web Service Composition Using Genetic Algorithm

Zhou Xiangbing[1,2], Ma Hongjiang[1], and Miao Fang[2]

[1] Department of Computer Science, Aba Teachers College, SichuanWenchuan, 623002, China
[2] School of Information Science and Technology, Chengdu University of Technology Chengdu, 610059, China
{3dsmaxmaya,mhj69,mfang}@163.com

Abstract. This paper presents an optimal QoS-based service composition approach to WSMO in helping service requests select services by considering two different contents: reasoning above the matching of semantic linking and GA-based QoS optimization of the WSMO service composition. The experimental results show that the approach being probed in this paper is not only efficient, but also increases availability and reliability of the WSMO service composition.

Keywords: WSMO, Web service, Quality of service, Service composition, Genetic algorithm.

1 Introduction

The semantic Web [1], where the semantic information uses machine-processed languages such as OWL(Web Ontology Language)[2], is considered to provide a number of advantages over the current version of the World Wide Web, which focuses on information automatic processing. Web service is an XML-based application program that enables communication between programs, databases and business functions [3]. The first appearance of the web service on World Wide Web is considered to change Web application schema and structure. As a combination of semantic Web and web service [4], Semantic Web service relies on formal semantic description (such as Description Logic) of web service functionality and interface to enable automated reasoning over web service compositions [5]. Description languages of the semantic web service turn to OWL-S(Ontology Web Language for Services) and WSMO(Web Service Modeling Ontology) for service functions and interfaces description; And the structure for WSMO service composition is derived when combined with VTA(Virtual Travel Agency). In this paper, we'd like to compose the web service by adopting WSMO, and construct a QoS-based WSMO service composition approach by using the GA (genetic algorithm). Our approach is to use both reasoning above the matching of semantic linking and GA-based QoS optimization so that the two perspectives can be included in our probing. The semantic linking matching builds WSMO service matching schema by semantic

A. Ghose et al. (Eds.): ICSOC 2012, LNCS 7759, pp. 127–139, 2013.

reasoning, while the GA-based QoS optimization builds a QoS-based composition model to obtain optimization results, between them is eventually achieved a higher quality service composition for requirements.

The Web service composition synthesizes individual and outsourced web services to turn out a new service; However, this still remains a very complex and challenging issue which defies the present human intelligence. Quality of service (short as QoS) concerns are becoming crucial to the global success in the web service based on computing paradigm [6]. Lots of QoS-based service composition approaches are in every detail discussed and summarized in the [5][6][7][8][12][13]; But most of the discussed approaches haven't made any mention of QoS-based WSMO service composition. Tomas Vitvar et al. [9] proposed WSMO-lite annotations for web services; MatthiasKlusch et al. [10] proposed A hybrid Semantic Web Service Matchmaker; Farshad Hakimpour et al. [11] proposed Semantic Web Service Composition in IRS-III: The Structured Approach , and so on, but the those researchers haven't yet found any good results for the WSMO service composition. In this case, it's of much significance for us to present an optimal approach to the QoS-based WSMO Web service composition by using genetic algorithm.

A WSMO service composition approach is proposed in this paper by considering the following two aspects: (1) an orchestration and a choreography which are called interface. Nevertheless, the interface is described by defining information exchange and matching reasoning methods. According to features of the interface, we have proposed the quantitative formulas for information exchange and matching reasoning; (2) a WSMO QoS model is presented by four aspects like information exchange, time, matching reasoning and price(cost). We define an optimal QoS-based WSMO Web service composition model, then employ genetic algorithm to obtain better results. The rest of the paper is organized as follows: Section 2 introduces our approach; Section 3 presents the experimental results; Section 4 comes to conclusions.

2 Our Approach

Three respects are dissertated and introduced into this section: semantic-based WSMO service composition approach discussed in section 2.1; a quality model of WSMO service presented in section 2.2; GA applied to optimize the quality model which obtains optimum results of composition in section 2.3.

2.1 Description of WSMO Service Composition Methods

The semantic web service is based on ontology, with no exception in WSMO. In other words, the semantic functions of web service can be implemented via WSMO, while these functions are normally described as input and output. Through $\mathcal{DL}\ \mathcal{EL}$ [14], the semantic recognition and conversion of ontology description are accomplished by the $\mathcal{DL}\ \mathcal{EL}$ inference engine [5][10][15]. At present, the main description languages of semantic web service are OWL-S profile and WSMO capability. The contents of WSMO service composition approaches are included in this paper as indicated in the following two aspects:

(1) Semantic linking is used to semantic web service composition;

(2) The quality of service (QoS) is constrained to meet the demands of different consumer(request) services.

Definition 1[5][15]. Give an ontology O which meets $<T, A>$ over $\mathcal{DL}$ $\mathcal{EL}$[14], namely, O: $<T, A>$, and T is a term Box, it is referred to as TBox(intentional knowledge). A is an assertion Box, it is known as ABox(extensional knowledge). TBox, which is usually used to describe different semantics in $\mathcal{DL}$ $\mathcal{EL}$, is recorded as TBox T.

In addition, the TBox in the **Definition 1** is described as **Example 1** with the following information: ***get request***, ***provide offer***, ***receive selection***, ***send confirmation***; ***flight request***, ***hotel request***, ***book flight***, ***book hotel***.

Example 1: A travelers books flight and hotel

$\forall$***get request***.(***flight request***, ***hotel request***)$\cap\exists$***provide offer.***(***book flight***, ***book hotel***)$\rightarrow$

$\exists$***get request*** . ***flight request*** $\vee$ $\exists$***get request .hotel request***

$\cup\forall$***provide offer. book flight*** $\vee$ $\forall$***provide offer. book hotel*** $\rightarrow$

receive selection.(***flight request, hotel request***) $\cup$***send confirmation***(***book flight, book hotel***)

Definition 2. WSMO service interface is described as:

$$WSI=<WS, Voc(In, Out, Share, Contr), \varepsilon(Voc), \kappa(\varepsilon)>.$$

Where,

WS denotes a semantic web service, *Voc* four models of information exchange: **input, output, sharing and control**, $\varepsilon(Voc)$ the message exchange state of ontology example definition, and $\kappa(\varepsilon)$ state transition structure (which is described as *if*(condition)…*then*(action)…) of layout (or choreography: an interaction between different services) and arrangement (or orchestration: a focus service for realizing the service function) [16]. T is used to describe and meet the requirements of O. At this time, the semantic web service is described as:

$$SWS=<T, O, WSI>$$

If $SWS=\{Sws_1, Sws_2, \ldots, Sws_i, Sws_j, \ldots, Sws_N\}$ is a given set, its input parameters and output parameters are described by Tbox T, and semantic linking is engendered through the $\mathcal{DL}$ $\mathcal{EL}$ comments. If each SWS has input(In_s_i) and output(Out_s_j) parameters, then:

$$\text{TBox } T\text{: } Sws_i\,(Out\_s_i \leftarrow WSI_i \rightarrow In\_s_i) \rightarrow$$

$$\text{TBox } T : Sws_j\,(Out\_s_j \leftarrow WSI_j \rightarrow In\_s_j)$$

In TBox T, SWS matching [5] is realized by using the similarity of various input and output parameters, the service composition is accordingly completed, and finally the satisfactory semantic linking will be accomplished as acquired in this paper.

Definition 3. A semantic linking of WSMO is defined as：

$$sl_{i,j}=< Sws_i, Sim_T\ (Out\_s_i \leftarrow WSI \rightarrow In\_s_j)\ , Sws_j>$$

If Out_s_i is the immediate successor of *SWS*, In_s_j is the immediate predecessor of *SWS*, that is to say, each *SWS* has input and output parameters, then these parameters are the information exchange foundation of *WSI*. Namely,

$$\forall_{Sws}, \exists(In\_s_i, Out\_s_i, WSI) \rightarrow Sws$$

And in TBox *T*, we choose $\pi = (\equiv, \sqsubseteq, \sqsupseteq, \cap, \perp, \sim)$ model to reason for the Sim_T matching calculation method:

Equivalence： $T \models Out\_s_i \xrightarrow[\equiv]{WSI} In\_s_j$

Plugin： $T \models Out\_s_i \xrightarrow[\sqsubseteq]{WSI} In\_s_j$

Subsume(inverse-plugin)： $T \models Out\_s_i \xrightarrow[\sqsupseteq]{WSI} In\_s_j$

Intersection： $T \models Out\_s_i \xrightarrow[\cap]{WSI} In\_s_j \sqsubseteq \perp, Out\_s_i \cap in\_s_j \neq \varnothing$

Disjunction： $T \models Out\_s_i \xrightarrow[\perp]{WSI} In\_s_j \sqsubseteq \perp, Out\_s_i \cap in\_s_j = \varnothing$

Fuzzy Operation: $T \models Out\_s_i \xrightarrow[\sim]{WSI} In\_s_j$

We assess the efficiency of $sl_{i,j}$ in the light of the following formula:

Give a set $|SWS|=N$, we can reason by the Sim_T matching reasoning rules, then the *Voc* information exchange reasoning efficiency in the *WSI* is defined as:

$$WSI(Voc_i, Voc_j) = \left\{ I_{(\frac{1}{2},1)} \left| \left(I_{(\frac{1}{2},1)} \rightarrow Voc_i \right) \prec \left(I_{(\frac{1}{2},1)} \rightarrow Voc_j \right) \right| \right\} \tag{2.1}$$

In (2.1), $\prec$ denotes the partial order operation of information exchange, | | value of partial order operation. When using *WSI* to realize information exchange, we choose different operation modes for matching reasoning according to different probabilities (1/2,1) of the *WSI* information exchange. Since there are four models(**input, output, sharing and control**) for *WSI* information exchange, its probability can be expressed as $I_{(\frac{1}{2},1)}$.

Where,

when *I*=1/2 is expressed as input and output which are necessary in the *Voc*, the *WSI* information exchange state is selected only one from **sharing** and **control**.

When *I*=1 is expressed as input and output which are also necessary in the *Voc*, the *WSI* information exchange state select both **sharing** and **control**.

SWS matching reasoning efficiency is defined as:

$$SWS(Out\_s_i, In\_s_j) = \left\{ \begin{array}{l} (Sws_i, Sws_j) \\ \left| (Out\_s_i \pi Sws_i) \prec (In\_s_j \pi Sws_j) \right| \neq \varnothing \end{array} \right\} \tag{2.2}$$

In (2.2), $\prec$ denotes the partial order operation of semantic linking, π is a group of reasoning of semantic linking.

According to (2.1) and (2.2), we can define the semantic linking efficiency, namely the efficiency of $sl_{i,j}$:

$$sl_{i,j}(Out\_s_i, In\_s_j) = \frac{\sum_{n\in N} |SWS(Out\_s_i, In\_s_j)|_n}{\sum_{n\in N} |WSI(Voc_i, Voc_j)|_n + \sum_{n\in N} |SWS(Out\_s_i, In\_s_j)|_n} \tag{2.3}$$

Example 2. Semantic linking efficiency

When N=10, (2.1), (2.2) ***and*** (2.3) ***values are computed following***:

WSI(***Voc$_i$***, ***Voc$_j$***)={0.1, 0.3, 0.5, 0.2, 0.4, 0.2}

SWS(***Out_s$_i$***, ***In_s$_j$***)={2, 4, 3, 2, 5, 1}

$$sl_{i,j}(Out\_s_i, In\_s_j) = \frac{2+4+3+2+5+1}{2+4+3+2+5+1+0.1+0.3+0.5+0.2+0.4+0.2} \approx 0.909$$

Definition 4. WSMO service composition

ASWSC: <SWS_N, max($sl_{i,j}$), QoS_E, *Tsk*>, the four respectively denote users' demands, maximum semantic web services(*N* numbers) semantic linking value, service quality evaluation(see Section 2.2) and the executed tasks.

Example 3. WSMO service composition

FWS: Flight web service(WSMO)
HWS:Hotel web service(WSMO)
VAT: Virtual Travel Agency
if FWSrequest then send(FWS, flightRequest)
Start(VTA, FWS)
if getFWSrequest then offerFWS
ComputeFWS(max(sl$_{i,j}$(Out_s$_i$, In_s$_j$)))
if FWSorder then comfirmation
if selection then book(FWS, flightBookingOrder)
Termination(VTA, FWS)
if flightorder then send(HWS, flightRequest)
Start(VTA, HWS)
if ***getHWSrequest then offerHWS***
ComputeHWS(max(sl$_{i,j}$(Out_s$_i$, In_s$_j$)))
if HWSorder then comfirmation
if get(selection, flightBookingConf) then book(HWS, hotelBookingOrder)
Termination(VTA, HWS)

2.2 Quality Model of WSMO Service Composition

Service quality is usually described by response time (including execution time and waiting time), cost, availability, reliability and credibility [3][5][6][7][8][12][17][18]. In this section, WSMO service quality is analyzed by transforming the availability, reliability and credibility into the quality of a *WSI* information exchange and matching reasoning. Then the QoS of WSMO is described as:

$$\begin{aligned} &\text{I: } QoS_E(q) = (Q_{rt(q)}, Q_{Wsi(q)}, Q_{sws(q)}, Q_{p(q)}) \\ &\text{II: } QoS_E(q) \rightarrow AF(\text{Sequence,Switch,Flow,Loop}) \end{aligned} \tag{2.4}$$

Table 1. Aggregation functions per execution path construct of QoS attributes

QoS attribute	Sequence	Switch	Flow	Loop
$Q_{rt(q)}$	$\sum_{i=1}^{n} q_{rt}^{i}$	$\sum_{i=1}^{n} S_{im} \cdot q_{rt}^{i}$	$\max\{(q_{rt}^{i})_{i\in\{1,\cdots,m\}}\}$	$\frac{1}{k} \cdot q_{rt}^{i}$
$Q_{Wsi(q)}$	$\prod_{i=1}^{n} q_{wsi(Vol_i,Vol_j)}^{i}$	$\sum_{i=1}^{n} S_{im} \cdot \log q_{wsi(Vol_i,Vol_j)}^{i}$	$\prod_{i=1}^{m} q_{wsi(Vol_i,Vol_j)}^{i}$	$q_{wsi(Vol_i,Vol_j)}^{i^{\frac{1}{k}}}$
$Q_{sws(q)}$	$\prod_{i=1}^{n} q_{sl_{i,j}(Out\_s_i,In\_s_j)}^{i}$	$\sum_{i=1}^{n} \left(S_{im} \cdot \log q_{sl_{i,j}(Out\_s_i,In\_s_j)}^{i} \right)$	$\prod_{i=1}^{m} q_{sl_{i,j}(Out\_s_i,In\_s_j)}^{i}$	$q_{sl_{i,j}(Out\_s_i,In\_s_j)}^{i^{\frac{1}{k}}}$
$Q_{p(q)}$	$\sum_{i=1}^{n} q_{p}^{i}$	$\sum_{i=1}^{n} S_{im} \cdot q_{p}^{i}$	$\sum_{i=1}^{m} q_{p}^{i}$	$\frac{1}{k} \cdot q_{p}^{i}$

In (2.4), the each element in the I respectively denote response time, information exchange rate, matching reasoning rate and price, whereas *AF* in the II describes aggregation functions per *epc*(execution path construct(Sequence, Switch, Flow, Loop))of QoS attributes [8]. The execution path construct is showed in **Table 1**(*k* denotes iterations). Different QoS values cannot be calculated and classified effectively, so we have to deal with these values by choosing normalized formula according to different QoS optimization values. So all the values in (2.4) are normalized by using the formula below:

$$q.value = \begin{cases} \dfrac{q.value - q.\min}{q.\max - q.\min} & \text{if } q.\max\text{-}q.\min \neq 0 \\ 1 & \text{if } q.\max\text{-}q.\min = 0 \end{cases} \tag{2.5}$$

$Q_{Wsi(q)}, Q_{Asws(q)}, Q_{p(q)}$

$$q.value = \begin{cases} \dfrac{q.\max - q.value}{q.\max - q.\min} & \text{if } q.\max\text{-}q.\min \neq 0 \\ 1 & \text{if } q.\max\text{-}q.\min = 0 \end{cases} \quad Q_{rt(q)} \tag{2.6}$$

Given that complete the task *j* is completed by SWS_N service request *i* in the service composition, if the choice of a task is made by service request *i*, we say the value is 1, otherwise 0, descried as follows:

$$S_{ij} = \begin{cases} 1 \\ 0 \end{cases} \tag{2.7}$$

In Table 1, if a task *j* is executed by *epc*(execution path construct) via using service request *i*, then different execution probability is defined in Table 2.

Table 2. Different execution probability in **Table 1**

QoS attribute	Sequence	Switch	Flow	Loop
$Q_{rt(q)}$	P_{rt}^1	P_{rt}^2	P_{rt}^3	P_{rt}^4
$Q_{Wsi(q)}$	P_{Wsi}^1	P_{Wsi}^2	P_{Wsi}^3	P_{Wsi}^4
$Q_{sws(q)}$	P_{sws}^1	P_{sws}^2	P_{sws}^3	P_{sws}^4
$Q_{p(q)}$	P_p^1	P_p^2	P_p^3	P_p^4

At this time, we get the normalization results of (2.4) according to (2.4)-(2.7) and Table 1 and Table 2:

$$\begin{aligned} Q_{rt(q)} &= P_{rt}^1 \cdot \sum\nolimits_{i=1}^{n} q_{rt}^i + P_{rt}^2 \cdot \sum\nolimits_{i=1}^{n} S \cdot_{im} q_{rt}^i + \\ & P_{rt}^3 \cdot \max\{\left(q_{rt}^i\right)_{i\in\{1,\cdots m\}}^{n_3}\} + P_{rt}^4 \cdot \left(\frac{1}{k} \cdot q_{rt}^i\right) \end{aligned} \tag{2.8}$$

$$\begin{aligned} Q_{Wsi(q)} &= \left(P_{Wsi}^1 \cdot \prod\nolimits_{i=1}^{n} q_{wsi(Vol_i,Vol_j)}^i\right) \\ & \cdot \left(P_{Wsi}^2 \cdot \sum\nolimits_{i=1}^{m} S_{im} \cdot \log q_{wsi(Vol_i,Vol_j)}^i\right) \\ & \cdot \left(P_{Wsi}^3 \cdot \prod\nolimits_{i=1}^{n} q_{wsi(Vol_i,Vol_j)}^i\right) \cdot \left(P_{Wsi}^4 \cdot \left(q_{wsi(Vol_i,Vol_j)}^{\frac{1}{i^k}}\right)\right) \end{aligned} \tag{2.9}$$

$$\begin{aligned} Q_{sws(q)} &= \left(P_{sws}^1 \cdot \prod\nolimits_{i=1}^{n} q_{sl_{i,j}(Out\_s_i,In\_s_j)}^i\right) \\ & \cdot \left(P_{sws}^2 \cdot \sum\nolimits_{i=1}^{m} S_{im} \cdot \log q_{sl_{i,j}(Out\_s_i,In\_s_j)}^i\right) \\ & \cdot \left(P_{sws}^3 \cdot \prod\nolimits_{i=1}^{n} q_{sl_{i,j}(Out\_s_i,In\_s_j)}^i\right) \cdot \left(P_{sws}^4 \cdot \left(q_{sl_{i,j}(Out\_s_i,In\_s_j)}^{\frac{1}{i^k}}\right)\right) \end{aligned} \tag{2.10}$$

$$\begin{aligned} Q_{rt(q)} &= P_p^1 \cdot \sum\nolimits_{i=1}^{n} q_p^i + P_p^2 \cdot \sum\nolimits_{i=1}^{n} S \cdot_{im} q_p^i + \\ & P_p^3 \cdot \sum\nolimits_{i=1}^{m} q_p^i + P_p^4 \cdot \left(\frac{1}{k} \cdot q_p^i\right) \end{aligned} \tag{2.11}$$

In order to make an operation between (2.9),(2.10) and (2.8) ,(2.11), we use logarithm function[6] to convert (2.9) and (2.10) into (2.12), (2.13).

$$
\begin{aligned}
Q_{Wsi(q)} &= \log\left(\begin{array}{l}\left(P^1_{Wsi}\cdot\prod_{i=1}^{n_1} q^i_{wsi(Vol_i,Vol_j)}\right)\\ \cdot\left(P^2_{Wsi}\cdot\sum_{i=1}^{m} S_{im}\cdot\log q^i_{wsi(Vol_i,Vol_j)}\right)\\ \cdot\left(P^3_{Wsi}\cdot\prod_{i=1}^{n_3} q^i_{wsi(Vol_i,Vol_j)}\right)\cdot\left(P^4_{Wsi}\cdot\left(q^{\frac{1}{i^k}}_{wsi(Vol_i,Vol_j)}\right)\right)\end{array}\right)\\
&= \log\left(\left(P^1_{Wsi}\cdot\sum_{i}^{n} q^i_{wsi(Vol_i,Vol_j)}\right)\right)+\log\left(\left(P^2_{Wsi}\cdot\sum_{i=1}^{m} S_{im}\cdot\log q^i_{wsi(Vol_i,Vol_j)}\right)\right)\\
&+\log\left(P^3_{Wsi}\cdot\sum_{i}^{m} q^i_{wsi(Vol_i,Vol_j)}\right)+\frac{1}{k}\cdot\log\left(P^4_{Wsi}\cdot\left(q^i_{wsi(Vol_i,Vol_j)}\right)\right)
\end{aligned}
\tag{2.12}
$$

$$
\begin{aligned}
Q_{sws(q)} &= \log\left(\begin{array}{l}\left(P^1_{sws}\cdot\prod_{i=1}^{n_1} q^i_{sl_{i,j}(Out\_s_i,In\_s_j)}\right)\\ \cdot\left(P^2_{sws}\cdot\sum_{i=1}^{m} S_{im}\cdot\log q^i_{sl_{i,j}(Out\_s_i,In\_s_j)}\right)\\ \cdot\left(P^3_{sws}\cdot\prod_{i=1}^{m} q^i_{sl_{i,j}(Out\_s_i,In\_s_j)}\right)\\ \cdot\left(P^4_{sws}\cdot\left(q^{\frac{1}{i^k}}_{sl_{i,j}(Out\_s_i,In\_s_j)}\right)\right)\end{array}\right)\\
&= \log\left(P^1_{sws}\cdot\sum_{i=1}^{n} q^i_{sl_{i,j}(Out\_s_i,In\_s_j)}\right)+\log\left(\left(P^2_{sws}\cdot\sum_{i=1}^{m} S_{im}\cdot\log q^i_{sl_{i,j}(Out\_s_i,In\_s_j)}\right)\right)\\
&+\log\left(P^3_{sws}\cdot\sum_{i=1}^{m} q^i_{sl_{i,j}(Out\_s_i,In\_s_j)}\right)+\frac{1}{k}\cdot\log\left(P^4_{sws}\cdot\left(q^i_{sl_{i,j}(Out\_s_i,In\_s_j)}\right)\right)
\end{aligned}
\tag{2.13}
$$

At this time, we can get the results of semantic WSMO service composition quality QoS_E according to (2.8),(2.11) ,(2.12) and (2.13), and set up an optimal model of QoS for WSMO service composition as illustrated in the following.

$$
\max F(S_{11}, \ldots, S_{nm}) = \max\left(\frac{w_{rt}Q_{rt(q)}+w_{Wsi}Q_{Wsi(q)}}{w_{sws}Q_{sws(q)}+w_p Q_{p(q)}}\cdot sl_{i,j}\right) \tag{2.14}
$$

Where,

$w_l \in [0, 1]$ is the weight assigned to lth quality of WSMO service composition and $\sum_{l\in\{rt,Wsi,sws,p\}} w_l = 1$.

constraints:

Response time: [1, 10000](ms)

Information exchange rate: [0.90, 0.99]

Matching reasoning rate: [0.90, 0.99]

Price: [1, 10000]

These constraints parameters are exploited in some research documents [19][20] where some correlated values are already set. Simulation tests are to be conducted in Section 3, we assume that all WSMO services run within 10000ms, and most of WSMO services are steady enough, then the information exchange rate and matching reasoning rate of WSMO services range from 0.90 to 0.99. Let us assume that it is reasonable for most of WSMO services and that the price is set in [21] on the basis of

the access cost of the WSMO services. It can be inferred that all these assumptions go for different applications.

2.3 A GA-Based Optimization for WSMO Service Composition

It is an NP-hard optimization problem [7] to acquire the best set of service for a WSMO composition to optimize a set of constraints. The solution to optimization composition of WSMO services is not functioning well, so we adopt a GA-based approach [5], since this approach supports constraints not only on QoS but also on quality of semantic linking, and requires the set of selected services as a solution to the maximization of a given objective *F* in the [5].

The optimal solution to GA-based approach is represented by *genotype*. Genotype is determined by simulating the evolution of an *initial population* within lots of generations. This kind of simulation eventually leads up to the survival of the *Fitness* individuals (WSMO service composition) satisfying some *constraints* and selection of WSMO service composition from the previous one.

(1) **Genotype.** The gene is defined by an array of integers. The number of genes in the array is equal to the number of *Tsk*s involved in the WSMO service composition. Each gene, in turn, contains an index to an array of candidate services for that *Tsk*, indicating a specific chosen service. Therefore, each composition, as a potential solution, can be encoded using this genotype.

(2) **Initial Population.** According to (1), the initial population consists of an initial set of WSMO service compositions.

(3) **Constraints.** They have to be met by (2) and *F*.

(4) **Fitness Function.** GA is running, constraints in the (3) have to met *F*. When constraints in the (3) haven't to met *F*, Lagrange penalty function is employed to renovate *F*[5].

$$f(S_{11},...,S_{nm}) = Y - \rho \cdot \frac{g}{mg} \cdot \sum_{k \in \{Q_{rt(q)}, Q_{Wsi(q)}, Q_{ows(q)}, Q_{p(q)}\}} \left(\frac{\Delta Q_k}{Q_k^{\max}(S_{11},...,S_{nm}) - Q_k^{\min}(S_{11},...,S_{nm})} \right)^2 \tag{2.15}$$

$$\Delta Q_k = \begin{cases} Q_k - Q_k^{\max} & \text{if } Q_k > Q_k^{\max} \\ 0 \quad \text{if } Q_k^{\min} \le Q_k \le Q_k^{\max} \\ Q_k^{\min} - Q_k & \text{if } Q_k > Q_k^{\max} \end{cases} \tag{2.16}$$

In (2.15), g is current genetic iterations, mg is maximum genetic iterations, $\rho \in [0,1]$ is penalty factor; $Q_k^{\min}$ is *k*th minimum value, $Q_k^{\max}$ is *k*th maximum value.

(5) **Genetic Operations.** They include mutation, crossover and selection. Mutation is random conversion value in an array(Genes), the mutation probability is recorded as P_{mutation}; Crossover is exchange of mutation genes and current genes, the crossover probability is recorded as $P_{\text{crossover}}$, and is defined as $P_{\text{crossover}} > P_{\text{mutation}}$; Selection expresses fitness value whether meet constraints, and select a optimized value as a current iterated value.

(6) **Stop Optimization.** When iterated value met constraints, the optimization value stop, and turn to other WSMO service composition.

3 Experimental Results

In this section, we analyze the performances of the approaches mentioned in Section 2 by way of:

(1) Comparison. In section 3.2, the evolution of the matching over the GA generations is considered as the default matching (Only on condition that service type deviations are explicitly granted if the goal derivations are allowed.) in the [10]. And the default matching is a GE (greed exhaustively) method, its generations are the same as GA.

(2) Efficiency. In section 3.3, we employ VTA(http://www.w3.org/Submission /WSMO) to evaluate efficiency(a feasible solution) of the WSMO composition between GA-based and IP-based by varying the number of tasks and candidate WSMO services. And each task is performed by selecting and invoking one or more service.

3.1 Experimental Configuration

We utilize Java programming language to complete these WSMO service composition and quality models and put Java-based programs in VTA and analyze[23] via ISR-III. Our GA extends the GPL library JGAP(http://jgap. sourceforge.net/), and the IP-based optimization problem is solved by running CPLEX[8]. All those experimental programs and tools run on an Inter Pentium (R) G630 2.7GHz 2.7GHz, and 2G RAM, Windows 7 and JDK6.0. , other experimental relation parameters see [5], and test software of VTA run on WSMO environment.

Composition with up to 30 tasks and 40 WSMO candidate services per task is discussed and tested in Sections 3.2 and 3.3, in which convincing results toward quality and feasible solution (Max. Fitness) for WSMO services are derived. The quality of the WSMO service composition is evaluated by means of percentage ($F(S_{11}, \ldots, S_{nm})$: Max. Fitness→Generation. Num) of the GA-based solution with respect to the multi-factors global optimum. The latter is procured by running the IP and exhaustive approaches with no time limit.

3.2 Evolution of WSMO Service Matching Efficiency

In this first experiment (showed in Fig.1), we focus on the benefits of WSMO service matching. Toward this end, we study the impact of matching between GA and GE according to [5][8][10].

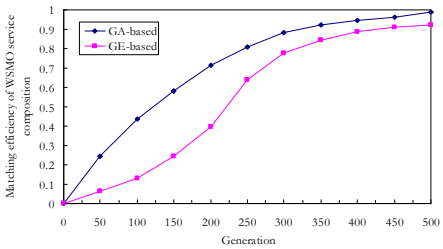

Fig. 1. WSMO service matching efficiency between GA-based and GE-based

To compare the different evolution of WSMO service matching efficiency, we present results between GA-based and GE-based. Results show that GA-based is able to find a good matching which meets the WSMO service composition.

3.3 Evolution of WSMO Service Composition

Fig.2 shows the evolution of the WSMO service composition efficiency (expressed via Fitness Function) over a lot of GA generations, and for different quality factors, we set weight values for assignment of numerous objectives for different number of tasks with 40 WSMO candidate services per task. Here, Fig.3 reports a WSMO service composition of 50 tasks wherein the number of candidate services varies from 1 to 400, and IP-based and GA-based approaches are compared in the Fig.3.

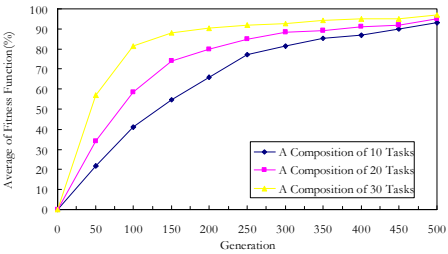

Fig. 2. Evolution of WSMO service composition

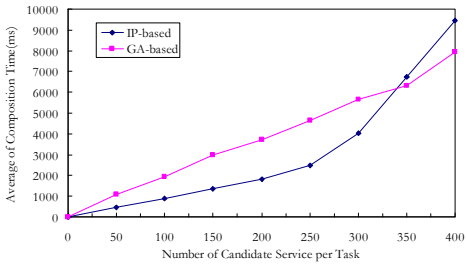

Fig. 3. Number of Candidate Service per Task

4 Conclusions and Future Work

The development of QoS-based and QoS-aware web service composition for subscribers is a popular issue in research since it is often viewed as a foundation of service-oriented computing. In addition, WSMO which is a semantic description method for web services has been applied for service computing. Therefore, we have presented an optimal approach to the WSMO service composition by using the matching model which allows specifying constraints on QoS for subscribers. An optimal QoS-based WSMO Web service composition approach is proposed in the paper in order to obtain higher information exchange rate and matching reasoning rate. The QoS-based WSMO fulfills the functional demands, for it not only helps subscribers to discover more web services, but also it satisfies the QoS constraints. Apart from that, a GA-based optimization is employed to evaluate the system performance of the WSMO service composition. The experimental results show that the QoS-based WSMO service composition approach to which an easy solution can be found is preferable to the IP-based.

In the next step, we plan to complete the framework for a particular application in WSMO domains. We will try to go deep into the QoS-based WSMO service composition like OWL-S-based approach and dynamic service composition, since they are currently beyond the reach of our research.

Acknowledgments. This work is supported by the Natural Sciences of Education and Science Office Bureau of Sichuan Province of China under Grant No. 2010JY0J41,11ZB152.

References

1. Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web. Scientific Am. 284(5), 34–43 (2001)
2. Smith, M.K., Welty, C., McGuinness, D.L.: OWL Web Ontology Language Guide. W3C Recommendation, W3C (2004)
3. Ko, J.M., Kim, C.O., Kwon, I.-H.: Quality-of-service oriented web service composition algorithm and planning architecture. Journal of System and Software 81(11), 2079–2090 (2008)
4. Mcllraith, S., Son, T.C., Zeng, H.: Semantic Web Services. IEEE Intelligent Systems 16(2), 46–53 (2001)
5. Lecue, F., Mehandjiev, N.: Seeking Quality of Web service composition in a semantic dimension. IEEE Transactions on Knowledge and Data Engineering 23(6), 942–959 (2011)
6. Huang, A.F.M., Lan, C.-W., Yang, S.J.H.: An optimal QoS-based Web service selection scheme. Information Science 179(19), 3309–3322 (2009)
7. Lin, C.-F., Sheu, R.-K., Chang, Y.-S., Yuan, S.-M.: A relaxable service selection algorithm for QoS-based web service composition. Information and Software Technology 53(12), 1370–1381 (2011)
8. Ardagna, D., Pernicl, B.: Adaptive Service Composition in Flexible Processed. IEEE Transactions on Software Engineering 33(6), 369–384 (2007)

9. Vitvar, T., Kopecký, J., Viskova, J., Fensel, D.: WSMO-Lite Annotations for Web Services. In: Bechhofer, S., Hauswirth, M., Hoffmann, J., Koubarakis, M. (eds.) ESWC 2008. LNCS, vol. 5021, pp. 674–689. Springer, Heidelberg (2008)
10. Klusch, M., Kaufer, F.: A hybrid Semantic Web Service Matchmake. Web Intelligence and Agent Systems: An International Journal 5(1-5), 1–20 (2008)
11. Hakimpour, F., Sell, D., Cabral, L., Domingue, J., Motta, E.: Semantic Web Service Composition in IRS-III: The Structured Approach. In: Seventh IEEE International Conference on E-Commerce Technology (CEC 2005), pp. 484–487 (2005)
12. Zeng, L., Benatallah, B., Ngu, A.H.H., Dumas, M., Kalagnanam, J., Chang, H.: Q-aware Middleware for web service composition. IEEE Trans. Software Eng. 30(5), 311–327 (2004)
13. Wang, H.H., Gibbins, N., Payne, T.R., et al.: A formal model of the Semantic Web Service Ontology (WSMO). Information Systems 37(1), 33–60 (2012)
14. Baader, F., Nutt, W.: The Description Logic Handbook: Theory, Implementation and Applications. Cambridge Univ. Press (2003)
15. Meditskos, G., Bassiliades, N.: Structural and Role-Oriented Web Service Discovery with Taxonomies in OWL-S. TEEE Trans. on Knowledge and Data Engineering 22(2), 278–289 (2010)
16. Kelle, U., Lara, R., Lausen, H., Fensel, D.: Semantic Web Services: Theory, Tools and Applications. Jorge Cardoso Press (2007)
17. Canfora, G., Di Penta, M., Esposito, R., et al.: A framework for QoS-aware binding and re-binding of composite web services. Journal of Systems and Software 81(10), 1754–1769 (2008)
18. Tsesmetzis, D., Roussaki, I., Sykas, E.: QoS-aware service evaluation and selection. European Journal of Operational Research 191(3), 1101–1112 (2008)
19. Zheng, Z., Lyu, M.R.: A QoS-aware fault tolerant middleware for dependable serivice composition. In: Proceeding of IEEE/IFIP International Conference on Dependable System & Networks, pp. 239–248 (2009)
20. Pastrana, J.L., Pimentel, E., Katrib, M.: Q-enabled and self-adaptive connectors for Web service composition and coordination. Computer Languages, Systems & Structures 37(1), 2–23 (2011)
21. Alonso, G., Casati, F., Kuno, H., Machiraju, V.: Web service Concepts, Architectures and Application. Springer, Heidelberg (2004)
22. Alrifai, M., Risse, T.: Combining Global Optimization with Local Selection for Efficient QoS-aware Service Composition. In: Proc. Int'l Conf. World Wide Web, pp. 881–890 (2009)
23. Demingue, J., Cabral, L., Galizia, S., et al.: IRS-III: A broker-based approach to semantic Web services. Journal of Web Semantics 6(2), 109–132 (2008)
24. Dang, N.C., Le, D.N., Quan, T.T., Nguyen, M.N.: Semantic Web Service Composition System Supporting Multiple Service Description Languages. In: Nguyen, N.T., Le, M.T., Świątek, J. (eds.) ACIIDS 2010. LNCS (LNAI), vol. 5990, pp. 390–398. Springer, Heidelberg (2010)
25. Vitvar, T., Mocan, A., Zaremba, M.: Formal Model for Semantic-Driven Service Execution. In: Sheth, A.P., Staab, S., Dean, M., Paolucci, M., Maynard, D., Finin, T., Thirunarayan, K. (eds.) ISWC 2008. LNCS, vol. 5318, pp. 567–582. Springer, Heidelberg (2008)
26. Lin, S.-Y., Lin, G.-T., Chao, K.-M., Lo, C.-C.: A Cost-Effective Planning Graph Approach for Large-Scale Web Service Composition. Mathematical Problems in Engineering 2012, Article ID 783476, 21 pages (2012), doi:10.1155/2012/783476