

Quantum Service Selection

Matthias Klusch

German Research Center for Artificial Intelligence
Multiagent System Group
Saarbrücken, Germany
klusch@dfki.de

René Schubotz

Digital Enterprise Research Institute
National University of Ireland, Galway
Galway, Ireland
rene.schubotz@deri.org

Abstract

Quantum service selection facilitates matchmaking of Web services and greatly benefits from quantum memory exponential in the number of its qubits as well as from computational speedup by an order of magnitude. Key idea of this work is the combination of Trugenberger's algorithms for associative quantum memories and the relaxation of semantic matchmaking to syntactical matchmaking within a graph space equipped with Hamming distance as its metric.

1 Introduction

With the increasing growth in popularity of Web services, Web service matchmaking becomes a significant challenge. Commonly, Web services are described by WSDL and published on UDDI registers. UDDI provides limited search facilities allowing only a keyword-based search.

Semantic Web services use OWL-S instead of WSDL to represent their capabilities. OWL-S provides a set of markup language constructs for an unambiguous, computer-interpretable service description. OWL-S markup will facilitate the automation of Web service tasks including automated Web service discovery, execution, interoperation, composition and execution monitoring. Classical software agents or search engines are enabled to discover suitable Web services via ontologies and reasoning methods.

In this paper, we develop a quantum computational method which assesses the similarity of Web services to achieve matchmaking. In particular, we present a conceptual model which combines quantum associative memories and the relaxation of semantic matchmaking to syntactical matchmaking within a graph space equipped with Hamming distance as its metric.

In section 2, we shortly comment on Web services, their representation in form of I/O signatures, and semantic sim-

ilarity. After providing a brief introduction to quantum computing in section 3, we detail on quantum associative memories in section 4, and give a quantum graph encoding scheme in section 5. The last part (section 6) introduces the concept of I/O signature graphs, and exposes our conceptual model of quantum service selection.

2 Web Services

Inspired by *Web service I/O matching* that exploits solely the parameter values of `hasInput` and `hasOutput`, a Web service S is subsequently specified by its *I/O signature*

$$S : IN_S \in T^m \rightarrow OUT_S \in T^n$$

where T is the terminology of a Web service matchmaker with concept subsumption hierarchy CT_T ; service inputs IN_S and outputs OUT_S range over the concepts $C \in T$.

An I/O signature provides the information necessary for a service-seeking agent to discover a suitable service. It includes a description of what is accomplished by the service in terms of required inputs and generated outputs concepts, but by no means instructs the agent how to use a discovered service.

Several degrees of semantic similarity between a service advertisement S and service request R have been identified in [4]. For a concept $C \in T$ we denote by $LSC(C)$ its immediate sub-concepts in CT_T , i.e. the set of least specific concepts of C .

EXACT A service S is said to be an exact match of a request R if the I/O signature of S perfectly matches with R with respect to equivalence of their semantics.

$$\forall IN_S \exists IN_R : IN_S \doteq IN_R \wedge \forall OUT_R \exists OUT_S : OUT_R \doteq OUT_S$$

PLUG-IN If a service S requires less input and is expected to return more specific output than specified in the request R , service S plugs into request R .

$$\forall IN_S \exists IN_R : IN_S \geq IN_R \wedge \forall OUT_R \exists OUT_S : OUT_S \in LSC(OUT_R)$$

SUBSUMES If a request R **subsumes** a service S , the set of eligible service advertisements is extended. Therefore, a subsumes match is weaker than a plug-in match with respect to the extent of the output's specificity.

$$\forall IN_S \exists IN_R : IN_S \geq IN_R \wedge \forall OUT_R \exists OUT_S : OUT_R \geq OUT_S$$

3 Quantum Computing In Very Brief

In the subsequent section we give some fundamental concepts of quantum computation. For a comprehensive introduction, we refer the reader to [5]. Quantum computation is built on the concept of the *qubit*. Any isolated physical 2-observable quantum system is appropriate to realize a single qubit. In mathematical terms, a qubit ψ is associated to its state space, a complex 2-dimensional Hilbert space $H_2 = \text{span}\{|0\rangle, |1\rangle\}$ with orthonormal computational standard basis. Any quantum state $|\psi\rangle$ of ψ is described by a *coherent superposition*, $|\psi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle$, $|\alpha_0|^2 + |\alpha_1|^2 = 1$. The state space $H_2^{\otimes n}$ of a physical system composed of n single qubits ψ_i is the n -folded *tensor product* of the state spaces of its n constituting qubits $H_2^{\otimes n} = \bigotimes^n H_2$. Such a system can be regarded as n -qubit register Ψ with 2^n computational basis states. If a state $|\Psi\rangle$ of a n -qubit register can be written as a product of its constituting qubits in the form $|\Psi\rangle = \bigotimes_i (\sum_j \alpha_{i,j} |j\rangle)$, then $|\Psi\rangle$ is called separable. Non-separable composite states are known as *entangled states*, allowing non-local effects of instantaneous state changes between spatially separated but entangled quantum states upon measurement. A *projective measurement* of Ψ is described by a set of pairwise orthogonal subspaces W_1, \dots, W_m satisfying $H_2^{\otimes n} = \bigoplus_{k=1}^m W_k$ and results in $j \in \{1, \dots, m\}$. Let $\{|\Phi_i^j\rangle\}$ define an orthonormal basis of subspace W_j , then the operator $P_j = \sum_{i=1}^{\dim(W_j)} |\Phi_i^j\rangle \langle \Phi_i^j|$ projects Ψ onto the subspace W_j . The probability of measuring $j \in \{1, \dots, m\}$ is given by $\langle \Psi | P_j | \Psi \rangle$. Time evolution of Ψ is described by *unitary transformations* in its state space $H_2^{\otimes n}$. Any non-measuring quantum operation is *inherently reversible* since any unitary transformation U has an inverse.

4 Quantum Associative Memories

Classical associative (or content-addressable) memories recall incomplete or noisy information but suffer from severe capacity shortages. Trugenberger [6] proposed a *model of a quantum memory* with optimal storage capacity implementing a retrieval mechanism for noisy or incomplete information. The number of binary patterns that can be stored in such a *quantum associative memory* is exponential in the number of qubits and thus optimal. In the following subsections we detail the pattern storage and pattern retrieval algorithm as given in [6].

4.1 Pattern Storage

A *quantum associative memory* for efficient storage and retrieval of $p \leq 2^n$ binary patterns p^i of n bits is naturally provided by the superposition $|M\rangle$ of n qubits

$$|M\rangle = \frac{1}{\sqrt{p}} \sum_{i=1}^p |p^i\rangle \quad (1)$$

Except for the unitary 1-qubit gate S^i

$$S^i = \begin{bmatrix} \sqrt{\frac{i-1}{i}} & \frac{1}{\sqrt{i}} \\ -\frac{1}{\sqrt{i}} & \sqrt{\frac{i-1}{i}} \end{bmatrix}$$

and the unitary controlled 2-qubit gate CS^i

$$CS^i = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \sqrt{\frac{i-1}{i}} & \frac{1}{\sqrt{i}} \\ 0 & 0 & -\frac{1}{\sqrt{i}} & \sqrt{\frac{i-1}{i}} \end{bmatrix}$$

the storage algorithm (algorithm 1) relies solely on the standard 1-qubit gates

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

as well as the standard 2-qubit gate

$$XOR = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

and its generalization $nXor$. For short hand notation, the subscripts of the gates refer to the control, input and output qubits on which they are applied.

In order to construct memory $|M\rangle$ from a simple initial state of n qubits, three qubit registers are needed: a pattern register $|p\rangle$ of n qubits into which the patterns p^i are subsequently fed, a utility register $|u\rangle$ of 2 qubits prepared in state $|01\rangle$, and a register $|m\rangle$ of n qubits to hold the memory, initially prepared in state $|0\rangle$.

$$|\Psi\rangle = |p\rangle \otimes |u\rangle \otimes |m\rangle$$

By the state of the utility register $|u\rangle$, the state of register $|\Psi\rangle$ is separated into two terms, one corresponding to the already stored patterns, and another for processing further patterns. The two terms are distinguished by the state of the second utility qubit u_2 : $|0\rangle$ for the stored patterns, and $|1\rangle$ for the processing term.

Algorithm 1 Creation of a quantum associative memory

Input: $p \leq 2^n$ binary n-bit patterns p^i

Output: $|M\rangle \in H_2^{\otimes n}$

Quantum Variables: $|\Psi\rangle \in H_2^{\otimes 2n+2}$, $|p\rangle, |m\rangle \in H_2^{\otimes n}$, $|u\rangle \in H_2^{\otimes 2}$

Classic Variables: $p, i \in \mathbb{N}$

```

1: Set  $|m\rangle = |\mathbf{0}\rangle$  and  $|u\rangle = |01\rangle$ 
2: for all  $i$  such that  $1 \leq i \leq p$  do
3:   Set  $|\Psi_0^i\rangle = |p^i\rangle \otimes |u\rangle \otimes |m\rangle$ 
4:   Set  $|\Psi_1^i\rangle = \prod_{j=1}^n 2XOR_{p_j^i u_2 m_j} |\Psi_0^i\rangle$ 
5:   Set  $|\Psi_2^i\rangle = \prod_{j=1}^n NOT_{m_j} XOR_{p_j^i m_j} |\Psi_1^i\rangle$ 
6:   Set  $|\Psi_3^i\rangle = nXOR_{m_1 \dots m_n u_1} |\Psi_2^i\rangle$ 
7:   Set  $|\Psi_4^i\rangle = CS_{u_1 u_2}^{p+1-i} |\Psi_3^i\rangle$ 
8:   Set  $|\Psi_5^i\rangle = nXOR_{m_1 \dots m_n u_1} |\Psi_4^i\rangle$ 
9:   Set  $|\Psi_6^i\rangle = \prod_{j=1}^n XOR_{p_j^i m_j} NOT_{m_j} |\Psi_5^i\rangle$ 
10:  Set  $|\Psi_0^{i+1}\rangle = |\Psi_7^i\rangle = \prod_{j=1}^n 2XOR_{p_j^i u_2 m_j} |\Psi_6^i\rangle$ 
11: end for
12: return  $|M\rangle = |m\rangle$ 

```

In every iteration of the main loop, a new pattern p^i is loaded into the pattern register (step 3), and the central operation of the storage algorithm applies in step 7. It separates out p^i , and sets the correct normalization factor.

Storage of a first pattern p^1 results in

$$|\Psi\rangle = \sqrt{\frac{1}{p}} |p^1, 00, p^1\rangle + \sqrt{\frac{p-1}{p}} |p^1, 01, \mathbf{0}\rangle$$

and $p-1$ further patterns can be processed as described above. The overall storage algorithm terminates with memory register $|m\rangle$ in state $|M\rangle$ as defined in equation 1.

4.2 Pattern Retrieval

The retrieval algorithm is inherently probabilistic and needs to be repeated until the measurement of a control qubit yields a result or a certain threshold is reached. In the former case, the output is determined by a amplitude distribution that is peaked around the stored patterns closest in Hamming distance to the input, while in the latter case, the input is not recognized.

Assume being provided with a binary input, that might be contained in a quantum associative memory $|M\rangle$. Since any retrieval from a quantum associative memory is a measurement and therefore reduces $|M\rangle$, the first step of retrieval produces a copy of $|M\rangle$. The no-cloning theorem prohibits perfect copies of quantum states, but a *sufficiently faithful copy* can be obtained with a probabilistic cloning machine [1].

The retrieval algorithm (algorithm 2) requires three registers: a n -qubit input register $|i\rangle$, a n -qubit memory register $|m\rangle$ containing the memory $|M\rangle$ and control register $|c\rangle$ initialized to $|0\rangle$. The full initial state (step 2) is thus given

Algorithm 2 Pattern retrieval from a quantum associative memory

Input: binary n-bit pattern i , quantum associative memory $|m\rangle = \frac{1}{\sqrt{p}} \sum_{k=1}^p |p^k\rangle$, threshold T

Output: pattern p_l such that $d_H(p_l, i) = \min \{d_H(p_k, i), 1 \leq k \leq p\}$

Quantum Variables: $|m\rangle \in H_2^{\otimes n}$, $|c\rangle \in H_2$

Classical Variables: k

```

1:  $k = 1$ 
2:  $|\Psi_0\rangle = |i, m, 0\rangle$ 
3: repeat
4:    $|\Psi_1\rangle = H_c |\Psi_0\rangle$ 
5:    $|\Psi_2\rangle = \prod_{k=1}^n NOT_{m_k} XOR_{i_k m_k} |\Psi_1\rangle$ 
6:    $|\Psi_3\rangle = e^{i \frac{\pi}{2n}} U_H |\Psi_2\rangle$ 
7:    $|\Psi_4\rangle = \prod_{k=1}^n XOR_{i_k m_k} NOT_{m_k} |\Psi_3\rangle$ 
8:    $|\Psi_5\rangle = H_c |\Psi_4\rangle$ 
9:   Perform projective measurement of control qubit
10:  if  $|c\rangle = |0\rangle$  then
11:     $|\Psi_6\rangle = \frac{P_0 |\Psi_5\rangle}{\sqrt{\langle \Psi_5 | P_0 | \Psi_5 \rangle}}$ 
12:    else
13:       $|\Psi_6\rangle = \frac{P_1 |\Psi_5\rangle}{\sqrt{\langle \Psi_5 | P_1 | \Psi_5 \rangle}}$ 
14:       $|\Psi_1\rangle = NOT_c |\Psi_6\rangle$ 
15:    end if
16:     $k = k + 1$ 
17:  until  $|c\rangle = |0\rangle$  or  $k = T$ 
18:  if  $|c\rangle = |0\rangle$  then
19:    input recognized, return measurement of memory register
20:  else
21:    input not recognized
22:  end if

```

by

$$|\Psi_0\rangle = \frac{1}{\sqrt{p}} \sum_{k=1}^p |i\rangle \otimes |p^k\rangle \otimes |0\rangle$$

The diagonal operator (D_H^i) computes the Hamming distance between an input $|i\rangle$ and a stored pattern $|p^k\rangle$ and is defined as follows

$$(D_H^i) = \sum_{k=1}^p (d_H(i, p^k)) |p^k\rangle \langle p^k|$$

In order to perform useful recognition, one accounts the Hamming distance of a memory element additive if $|c\rangle = |0\rangle$ and subtractive if $|c\rangle = |1\rangle$. This is done using

$$\begin{aligned} U_H &= (D_H^i)_m \otimes Z_c \\ &= (D_H^i)_m \otimes \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}_c \end{aligned}$$

Since operator U_H is not unitary, one uses the unitary matrix $e^{i \frac{\pi}{2n}} U_H$ in step 6. By applying the Hadamard gate to the control qubit in step 8, the quantum memory evolves to

$$\begin{aligned} |\Psi_5\rangle &= \sum_{k=1}^p \frac{1}{\sqrt{p}} \cos\left(\frac{\pi}{2n} d_H(i, p^k)\right) |i, p^k, 0\rangle \\ &+ \sum_{k=1}^p \frac{1}{\sqrt{p}} \sin\left(\frac{\pi}{2n} d_H(i, p^k)\right) |i, p^k, 1\rangle \end{aligned}$$

In order to determine if input i was recognized, one needs to perform a *projective measurement* of the control qubit $|c\rangle$ in step 9. If $|c\rangle = |0\rangle$, the input was recognized and a measurement of the memory register $|m\rangle$ will retrieve a pattern close to it in terms of Hamming distance. The probability for $|c\rangle$ to be in state $|0\rangle$ after the first iteration is given by

$$P(|c\rangle = |0\rangle) = \langle \Psi_5 | P_0 | \Psi_5 \rangle = \frac{1}{p} \sum_{k=1}^p \cos^2\left(\frac{\pi}{2n} d_H(i, p^k)\right)$$

with projector P_0 defined by

$$P_0 = \sum_{k=0}^{\frac{p}{2}-1} |2k\rangle \langle 2k|$$

Hence, the probability of measuring pattern p^k is given by

$$P(|m\rangle = |p^k\rangle) = \frac{\cos^2\left(\frac{\pi}{2n} d_H(i, p^k)\right)}{\sum_{k=1}^p \cos^2\left(\frac{\pi}{2n} d_H(i, p^k)\right)}$$

If $|c\rangle = |1\rangle$, the input was not recognized and the retrieval algorithm needs to be repeated until a threshold T is reached. The probability for $|c\rangle$ to be in state $|1\rangle$ is given by

$$P(|c\rangle = |1\rangle) = \langle \Psi_5 | P_1 | \Psi_5 \rangle = \frac{1}{p} \sum_{k=1}^p \sin^2\left(\frac{\pi}{2n} d_H(i, p^k)\right)$$

with projector P_1 defined by

$$P_1 = \sum_{k=0}^{\frac{p}{2}-1} |2k+1\rangle \langle 2k+1|$$

The post-measurement state for unrecognized input i is given by

$$\begin{aligned} |\Psi_6\rangle &= \frac{P_1 | \Psi_5 \rangle}{\sqrt{\langle \Psi_5 | P_1 | \Psi_5 \rangle}} \\ &= \frac{1}{\sqrt{\sum_{k=1}^p \alpha^2}} \sum_{k=1}^p \alpha |i, p^k, 1\rangle \end{aligned}$$

with

$$\alpha = \sin\left(\frac{\pi}{2n} d_H(i, p^k)\right)$$

The post-measurement state goes subsequently through the retrieval procedure until $|c\rangle = |0\rangle$ or a predefined threshold T is reached. This threshold determines the number of repetitions of the retrieval algorithm and thus the *recognition* and *identification* efficiency.

If T repetitions of the algorithm all lead to a measurement $|c\rangle = |1\rangle$, the input i is classified as *non-recognized*. If one measures $|c\rangle = |0\rangle$ before the threshold T is reached, the input i is *recognized* and adjacent measurement of the memory register $|m\rangle$ *identifies* the closest pattern.

Algorithm 3 Building a I/O signature graph from Web service advertisement S

Input: inputs $IN_S = \{(i, C_i)\}$, outputs $OUT_S = \{(i, C_i)\}$, threshold d
Output: I/O signature graph $G = (T, E)$

```

Set  $E = \emptyset$ 
for  $(i, C_i) \in IN_S$  do
    Set  $E = E \cup (C_i, Input_i)$ 
end for
for  $(i, C_i) \in OUT_S$  do
    Set  $E = E \cup (C_i, Output_i)$ 
end for
for  $(i, C_i) \in IN_S$  do
    Set  $E = E \cup (C_i, LGC(C_i))$ 
    Set  $IN_S = IN_S \cup (i, LGC(C_i))$ 
end for
for  $(i, C_i) \in OUT_S$  do
    if  $d \geq 0$  then
        Set  $E = E \cup (Output_i, C_i)$ 
    end if
    Set  $E = E \cup (C_i, LGC(C_i))$ 
    Set  $OUT_S = OUT_S \cup (i, LGC(C_i))$ 
    Set  $d = d - 1$ 
end for
return  $G = (T, E)$ 

```

According to Trugenberger, the efficiency of algorithm 2 depends on the threshold T determining recognition and on the distribution of stored patterns determining identification. The overall complexity is bounded from above by $O(\sqrt{p})$, basically the same result that can be obtained by Grover's search algorithm [2].

5 Graph Encoding and Quantum Graphs

By considering a trivial enumeration scheme for classical directed graphs, we now present a way to construct *quantum graphs*, i.e. superpositions of classical directed graphs.

Vertices of a directed graph G can be labelled with natural numbers ranging from $0, \dots, n-1$, i.e. vertices reside in a *space of vertices* V of size n . An edge $e = (p, q) \in V \times V$ over the space of vertices can be represented by

$$\langle e \rangle = pn + q$$

resulting in a *space of edges* E of size n^2 .

Given that, graphs over a space E of edges are enumerated using n^2 -bit words

$$\langle G \rangle = \sum_{e \in E_G} 2^{\langle e \rangle}$$

and reside in the *space of graphs* with size 2^{n^2} . Each element of the space of graphs corresponds to a classical directed graph with vertices and edges in the underlying spaces.

Algorithm 4 Building a I/O signature graph from Web service request R

Input: inputs $IN_R = \{(i, C_i)\}$, outputs $OUT_R = \{(i, C_i)\}$, threshold d
Output: I/O signature graph $G = (T, E)$

```

Set  $E = \emptyset$ 
for  $(i, C_i) \in IN_R$  do
  Set  $E = E \cup (C_i, Input_i)$ 
end for
for  $(i, C_i) \in OUT_R$  do
  Set  $E = E \cup (C_i, Output_i)$ 
end for
for  $(i, C_i) \in IN_R$  do
  if  $d \geq 0$  then
    Set  $E = E \cup (Input_i, C_i)$ 
  end if
  Set  $E = E \cup (C_i, LGC(C_i))$ 
  Set  $IN_R = IN_R \cup (i, LGC(C_i))$ 
  Set  $d = d - 1$ 
end for
for  $(i, C_i) \in OUT_R$  do
  Set  $E = E \cup (C_i, LGC(C_i))$ 
  Set  $OUT_R = OUT_R \cup (i, LGC(C_i))$ 
end for
return  $G = (T, E)$ 

```

The distance between two graphs G and H is naturally given by the number of edges that one contains that the other does not contain. Considering the proposed graph encoding, the distance between G and H is defined as the *Hamming distance*

$$d(G, H) = d_H(\langle G \rangle, \langle H \rangle) \quad (2)$$

A n^2 -qubit *quantum graph* $|G\rangle$ can be thought of as a superposition of classical directed graphs G_1, \dots, G_k from a space of graphs G with size 2^{n^2} and is given by

$$|G\rangle = \sum_{i=1}^k \alpha_i |G_i\rangle, \quad \sum_{i=1}^k \alpha_i^2 = 1$$

Hence, a quantum graph can store up to 2^{n^2} classical directed graphs using only n^2 qubits.

6 Quantum Service Selection

I/O signature graphs are canonical representations of Web services specified by their I/O signatures. In terms of graph theory, an I/O signature graph is a directed subgraph of a matchmaker's concept subsumption hierarchy CT_T with additional information regarding the service input and output parameters. Figure 1 provides simple examples of a concept subsumption hierarchy and an I/O signature graph.

Having introduced the basic concepts of our model of quantum service selection, i.e. quantum associative memories, quantum graphs and I/O signature graphs, the remainder of this section explicates the quantum storage and

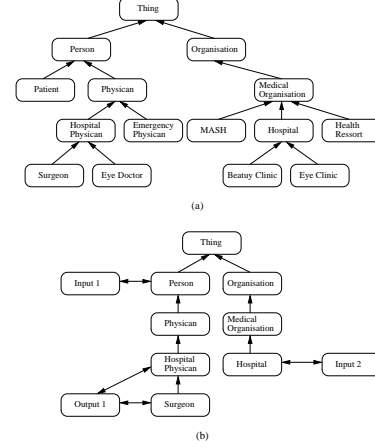


Figure 1. (a) Example of concept subsumption hierarchy (b) I/O signature graph for service advertisement $S : Person \times Hospital \rightarrow Surgeon$

matchmaking of Web services and illustrates the matchmaking procedure by an exemplaric Web service request.

6.1 Web Service Storage

Given the I/O signature $S : IN_{adv} \in T^m \rightarrow OUT_{adv} \in T^n$ of Web service advertisement adv , its input and output concepts IN_{adv}, OUT_{adv} are semantically integrated into a matchmaker's concept subsumption hierarchy CT_T using standard methods of ontology mapping. The corresponding I/O signature graph G_{adv} for Web service advertisement adv is constructed using algorithm 3 in time

$$O((m+n)depth(CT_T))$$

In order to store the Web service advertisement adv within a matchmaker's quantum associative memory, the I/O signature graph G_{adv} is binary encoding in time

$$O(m+n)$$

according to the encoding scheme provided in section 5.

The resulting binary pattern $\langle G_{adv} \rangle$ representing I/O signature graph G_{adv} is inserted into the provided quantum associative memory using algorithm 1 terminating with memory register $|M\rangle$ in state

$$|M\rangle = \frac{1}{\sqrt{N}} \sum_{i=1}^N |\langle G_{adv_i} \rangle\rangle$$

The overall storage procedure for Web service advertisements is given by algorithm 5.

Algorithm 5 Quantum Storage of Service Advertisement

Constant: threshold d

Input: service advertisement $adv : IN_{adv} \rightarrow OUT_{adv}$

Output: void

1. Classify $C \in IN_{adv}, C \in OUT_{adv}$ into local taxonomy
 2. Compute I/O signature graph G_{adv} for adv using algorithm 3
 3. Encode G_{adv} using encoding scheme from section 5 resulting in binary word $\langle G_{adv} \rangle$
 4. Store $\langle G_{adv} \rangle$ in quantum associative memory using algorithm 1
-

6.2 Web Service Matchmaking

Given the I/O signature $R : IN_{req} \in T^m \rightarrow OUT_{req} \in T^n$ of Web service request req , its input and output concepts IN_{req}, OUT_{req} are semantically integrated into a matchmaker's concept subsumption hierarchy CT_T using standard methods of ontology mapping. The corresponding I/O signature graph G_{req} for Web service request req is constructed using algorithm 4 in time

$$O((m+n)\text{depth}(CT_T))$$

Web service request I/O signature graph G_{req} is binary encoded in time

$$O(m+n)$$

and the corresponding binary word $\langle G_{req} \rangle$ is provided as input for retrieval algorithm 2 of a matchmaker's associative quantum memory.

Assuming a matchmaker's quantum associative memory register $|M\rangle$ in state

$$|M\rangle = \frac{1}{\sqrt{N}} \sum_{i=1}^N |\langle G_{adv_i} \rangle\rangle$$

the retrieval algorithm 2 yields the encoded Web service advertisement $\langle G_{adv_m} \rangle$ under the condition

$$d_H(\langle G_{adv_m} \rangle, \langle G_{req} \rangle) = \min_{i=1}^N \{d_H(\langle G_{adv_i} \rangle, \langle G_{req} \rangle)\}$$

with highest probability in time

$$O(\sqrt{N})$$

In terms of semantic similarity as defined in section 2, one concludes that the most relevant service advertisement with respect to the graph distance function (equation 2) is expected to be EXACT matching. The overall matchmaking procedure is given in algorithm 6.

6.3 Example

For the sake of illustration, we give an exemplaric service request

$$R : Person \times Hospital \rightarrow Physican$$

Algorithm 6 Quantum Matchmaking for Service Request

Constant: threshold d

Input: service request $req : IN_{req} \rightarrow OUT_{req}$

Output: service proposal $match$

1. Classify $C \in IN_{req}, C \in OUT_{req}$ into local taxonomy
 2. Compute I/O signature graph G_{req} for req using algorithm 4
 3. Encode G_{req} using encoding scheme from section 5 resulting in binary word $\langle G_{req} \rangle$
 4. Retrieve encoded matching service $\langle G_{match} \rangle$ for req from quantum associative memory using algorithm 2
- return** Decoded matching service $match$
-

and service advertisements of different matching degrees.

$$E : Person \times Hospital \rightarrow Physican$$

$$P : Person \times Organisation \rightarrow HospitalPhysican$$

$$S : Person \times Organisation \rightarrow Surgeon$$

Input and output concepts of request and advertisements range over the taxonomy CT drafted in figure 1(a). Figure 2 shows the respective I/O signature graphs, constructed with threshold $d = 1$ using algorithms 3, 4.

For graph encoding according to section 5, terminology T of CT is enumerated in a breadth-first manner, followed by auxiliary concepts for input and output annotation. The resulting binary I/O signature graph encodings are as follows.

$$\begin{aligned} \langle R \rangle &= \sum_{i \in \{15, 18, 22, 36, 73, 89, 91, 106, 167, 178, 270, 271, 293, 297, 319\}} 2^i \\ \langle E \rangle &= \sum_{i \in \{18, 33, 35, 36, 73, 89, 91, 167, 178, 271, 297, 307, 310\}} 2^i \\ \langle P \rangle &= \sum_{i \in \{18, 33, 36, 52, 73, 89, 112, 125, 271, 290, 310, 312\}} 2^i \\ \langle S \rangle &= \sum_{i \in \{18, 33, 36, 52, 73, 112, 125, 204, 215, 271, 290, 312, 317\}} 2^i \end{aligned}$$

Given the binary encoded I/O signature graphs, we can easily calculate graph space distances between service request R and the given service advertisements.

$$d(R, E) = d_H(\langle R \rangle, \langle E \rangle) = 4$$

$$d(R, P) = d_H(\langle R \rangle, \langle P \rangle) = 7$$

$$d(R, S) = d_H(\langle R \rangle, \langle S \rangle) = 9$$

Considering the properties of quantum associative memories, the memory's amplitude distribution is peaked around the stored service advertisement I/O signature graph ($E : Person \times Hospital \rightarrow Physican$) closest in Hamming distance to the service request I/O signature graph. In terms of semantic similarity, algorithm 2 returns an exact matching service with highest probability.

Furthermore, I/O signature graphs created with threshold $d > 0$ enforce quantum associative memories to prefer

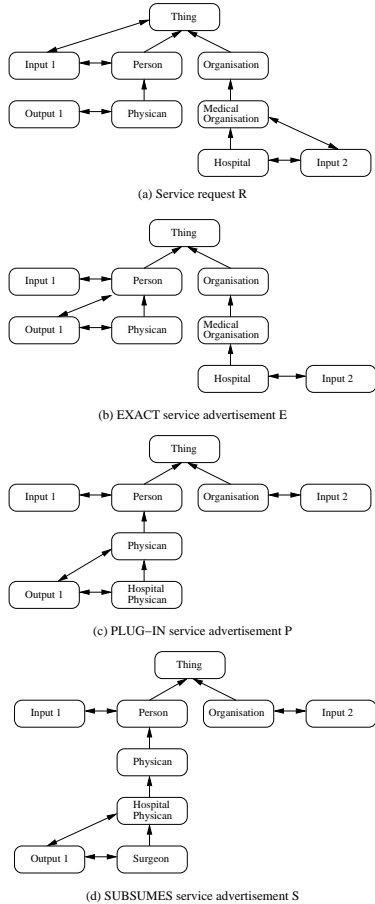


Figure 2. I/O signature graphs with threshold $d = 1$ for service request R and service advertisements E , P , and S

PLUG-IN over a SUBSUMES service advertisements under the condition that both have considerably similar input concepts.

7 Related Work

To the best of our knowledge, the presented paper is unique as there are no alternative conceptions of quantum Web service matchmaking available yet.

Our work combines related work from quantum computing and Web service matchmaking, and builds upon existing work on QC agents [3]. In particular, we exploit quantum associative memories proposed in [6], and several degrees of Web service semantic similarity identified in [4].

Furthermore, we give significant indication of quantum matchmakers outperforming existing classical approaches to Web services matchmaking based both on keyword search and ontologies and reasoning methods by an order

of magnitude.

8 Conclusions

In this paper, we presented a first quantum computational approach to Web service matchmaking.

Since several studies provide evidence in favor of the proposition that building semantic Web service matchmakers purely on reasoning methods may be insufficient, we relaxed semantic matchmaking to syntactical matchmaking within a graph space equipped with Hamming distance as its metric.

Given N service advertisements, the proposed service selection procedure of complexity $O(\sqrt{N})$ outperforms classical matchmaking approaches with complexity $O(N \otimes m)$, where m is the complexity of matchmaking. The theoretical performance of our approach over classical matchmaking provides strong evidence for the expected significant speed up of service matchmaking processes on the quantum internet.

However, the physical realization and computational simulation of quantum Web service matchmaking remains one open problem since current quantum computers are restricted to a few qubits and available quantum simulators severely suffer from computational complexity.

References

- [1] L.-M. Duan and G.-C. Guo. Probabilistic cloning and identification of linearly independent quantum states. *Phys. Rev. Lett.*, 80(22):4999–5002, Jun 1998.
- [2] L. K. Grover. From schrödinger’s equation to the quantum search algorithm. *American Journal of Physics*, 69(7):769–777, 2001.
- [3] M. Klusch. Toward quantum computational agents. In M. Nickles, M. Rovatsos, and G. Weiß, editors, *Agents and Computational Autonomy*, volume 2969 of *Lecture Notes in Computer Science*, pages 170–186. Springer, 2003.
- [4] M. Klusch, B. Fries, and K. Sycara. Automated semantic web service discovery with owls-mx. In *AAMAS ’06: Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, pages 915–922, New York, NY, USA, 2006. ACM Press.
- [5] M. A. Nielsen and I. L. Chuang. *Quantum computation and quantum information*. Cambridge Univ. Press, Cambridge, 2000.
- [6] C. A. Trugenberger. Probabilistic quantum memories. *Phys. Rev. Lett.*, 87(6):067901, Jul 2001.