ISSN : 2278 – 1021

International Journal of Advanced Research in Computer and Communication Engineering,
Vol. 1, Issue 7, September 2012

QoS Based Web Services Composition using Ant Colony Optimization: Mobile Agent Approach

Sunil R Dhore, Prof. Dr M U Kharat

Associate Professor & Member IEEE, Dept. Of CSE, Army Institute of Technology, Pune, India

Professor, Dept. Of CSE, MET's Institute of Engineering, Bhujbal Knowledge City, Nashik, M.S. India

Abstract: Web Services are based on distributed technology and provide standard means of interoperating between different software applications across and within organizational boundaries with the use of XML. Web Services technologies allow interaction between applications. Sometimes a single service given alone does not meet user's needs. In this case, it is necessary to compose several services in order to achieve the user's goal. In web service composition, a repository of services is given in which input and output parameters of each service are annotated with a concept from ontology. Given request is defined by a set of known input concepts and a set of wanted output concepts. Semantic composer should find a sequence of services, called composition that satisfies the wanted output concept. In this paper, a semantic composer based on ant colony optimization method is proposed. The proposed composer can find optimal composition length in each challenge set in a reasonable time. The result of the suggested composition can be compare with the best result reported in Web service consortium (WSC).

The novelty of this work lies with our multi-objective optimal-path selection modelling for QoS-based dynamic web service composition and a new version of ACO algorithm that is proposed to solve this multi-objective optimization problem.

Keywords: Web Service, Web Service Composition, Quality of Service, Multi-Agent System, Service-oriented architecture, Multi Objective Ant Colony Optimization

I. INTRODUCTION

Web Services as technology based on open XML standards like SOAP, WSDL, and UDDI are widely used for integration purposes within enterprises. Beyond this, Web Services have the potential to be composed to cross organizational workflows. Due to their loosely-coupled nature Web Services hosted by external providers can be integrated at runtime. This vision aims at dynamic ad-hoc collaborations between different business partners. With the increasing number of Web Services with similar or identical functionality, the non-functional properties of a Web Service will become more and more important. Besides the costs for using Web Services, the Quality of Service (QoS) attributes (e.g. availability, response time, and throughput) are subsumed as non-functional attributes. Considering those non-functional properties is crucial for companies to meet the requirements of customers.

As a consequence, the QoS has to be explicitly managed at the designing phase of a Web Service composition as well as during its execution at runtime. Focusing on the QoS-aware Web Service execution we have propose the proxy architecture AWSQoSX (Agent based Web Services Quality of Service Architectural Extension) that supports late binding of Web Services at runtime as well as dedicated accounting and monitoring mechanisms. In this proposal, the focus is on QoS-aware Web service composition. We define QoS-aware Web Service composition as the selection of Web Services maximizing the QoS of the overall Web Service composition, taking into account preferences and constraints defined by the user. For this, a utility function maximizing the overall QoS subject to QoS constraints is introduced. This leads to an optimization problem that is NP-hard.

In this paper, we propose Ant Colony Optimization method and Mobile agent approach for solving the QoS-aware Web Service composition problem. This paper then introduces four basic objective functions. The advantage of proposed approach is to recognize the feasibility of the composition process at any point of execution and produce better throughput and less consumption of memory to select composition of web services.

AIM: To design and analyze Multi-objective agent based Ant colony optimization QoS Model and service decomposition

OBJECTIVE: QoS performance comparison and analysis.

II LITERATURE SURVEY

Reference [01] discusses the Quality of Service (QoS)-aware composition of web Services. The work is based on the assumption that for each task in a workflow a set of alternative Web Services with similar functionality is available and that these Web Services have different QoS parameters and costs. This leads to the general optimization problem of how to select Web Services for each task so that the overall QoS and cost requirements of the composition are satisfied.

In Reference[02] run-time binding of web services has been recently put forward in order to support rapid and dynamic web service compositions. As with the growing number of alternative web services that provide the same functionality but differ in quality parameters, the service composition becomes a decision problem on which component services should be selected such that user's end-to-end QoS requirements (e.g. availability, response time) and preferences (e.g. price) are satisfied. The proposed solution consists of two steps: first, they use mixed integer programming (MIP) to find the optimal decomposition of global QoS constraints into local constraints. Second, proposer use distributed local selection to find the best web services that satisfy these local constraints.

Reference [04] In this reference the researcher describe their solution for pro-active Web services selection and composition. In particular, they consider a logic-based method for services composition and marketplace-based system architecture supporting agent communication, negotiation and semantically reasoning in the composition process.

Further in Reference [05] The proposed approach has a new multi-layer Web Services composition model based on Multi-Agent System. They propose a different self-adaptive mechanisms corresponding to different environment's evolutions to improve the reliability of Web Services composition efficiently. Also author introduce an algorithm for dynamic approach to select the best composition. This composition is selected based on the quality and the compose-ability of participated services. Advantage of the proposed approach is to recognize the feasibility of the composition process at any point of execution and produce better throughput and a less consumption of memory to select composition of services dynamically.

To use web service composition in service oriented architecture **Reference[06]** explains Service-Oriented Architecture (SOA) as a flexible framework for service composition. The problem is modelled in two ways: the combinatorial model and the graph model. The combinatorial model defines the problem as a multi dimension multi-choice 0-1 knapsack problem (MMKP). The graph model defines the problem as a multi-constraint optimal path (MCOP) problem. Efficient heuristic algorithms for service processes of different composition structures are presented in this article and their performances are studied by simulations.

In **Reference[07]** authors present a QoS broker based architecture for web services. The main goal of the architecture is to support the client in selecting web services based on his/her required QoS. To achieve this goal, they propose a two-phase verification technique that is performed by a third party broker. The first phase consists of syntactic and semantic verification of the service interface description including the QoS parameters description. The second phase consists of applying a measurement technique to compute the QoS metrics stated in the service interface and compares their values with the claimed one. This is used to verify the conformity of a web service from the QoS point of view (QoS testing).

In the **Reference [11]** the authors explains about "Performance Analysis of Mobile Agents" says Mobile-agent performance is a critical issue for mobile-agent based systems and their feasibility. They have utilized discrete-event simulation technique to build a performance evaluation scheme for generic mobile-agent systems.

To prove the use of Ant colony optimization in service oriented architecture we referred **Reference [12]** which says Service composition algorithms play an important role in selecting services from different providers to reach desirable QoS levels according to the performance requirements of composite services, and improve customer satisfaction. This paper proposes a novel QoS-based dynamic service composition technique for web services with Ant Colony Optimization (ACO) in an optimization approach.

Reference[18] on "Mobile Agent System for Web Services Integration in Pervasive Networks" elaborate

ISSN : 2278 – 1021

International Journal of Advanced Research in Computer and Communication Engineering,
Vol. 1, Issue 7, September 2012

about the integration of mobile agents for web services, the study in this paper adopts the mobile agent Technology in response to this problem and presents a mobile agent system for Web Services integration

The need of the agent based technology in internet programming is discuss in **Reference[22]** has titled "Mobile Agents and the Future of the Internet" and to justify this, authors presents the use of mobile agents and code by mentioning that Use of the Internet has exploded in recent years with the appearance of the World-Wide Web. In this paper, author show how current technological trends may lead to a system based substantially on mobile code, and in many cases, mobile agents. Author discusses several technical and non-technical hurdles along the path to that eventuality. It seems likely that, within a few years, nearly all major Internet sites will be capable of hosting and willing to host some form of mobile code or mobile agents.

Reference[36] Explains about Web Service Negotiation is a technology that autonomous entities reach agreement and meet both needs through negotiation. Web Service negotiation mechanism and coordinated-negotiation architecture on this basis have become hot issues of current research. The Coordinated-Negotiation Architecture of Web Service Composition based on Agent is a relative popular one in current architectures. However, there are some problems of it: unreasonable use of time and data link resources in the condition of multi-negotiation concurrency, which results in inefficiency of negotiation; lack of effective processing when confronting negotiation failure.

Reference[39] discovering web services using keyword based search techniques offered by existing UDDI API may not yield results that are tailored the client needs. When discovering the web services , clients look for those that meet their requirements, primarily the overall functionality and QoS. Standard such as UDDI, WSDL and SOAP have the potential of providing QoS aware discovery, however there is technical challenges associated with existing standard. This is overcome by web service relevancy function used for measuring the relevance of ranking of particular web service.

Finally in Reference[41] web service composition using agent and ACO is become concrete because it says, Dynamic Web Service composition is a key technology for building the service-oriented, loose coupling and integrated applications. In this paper, author present a novel global QoS optimizing and multi-objective Web Services selection algorithm based on Multi-objective Ant Colony Optimization (MOACO) for the Dynamic Web Service composition.

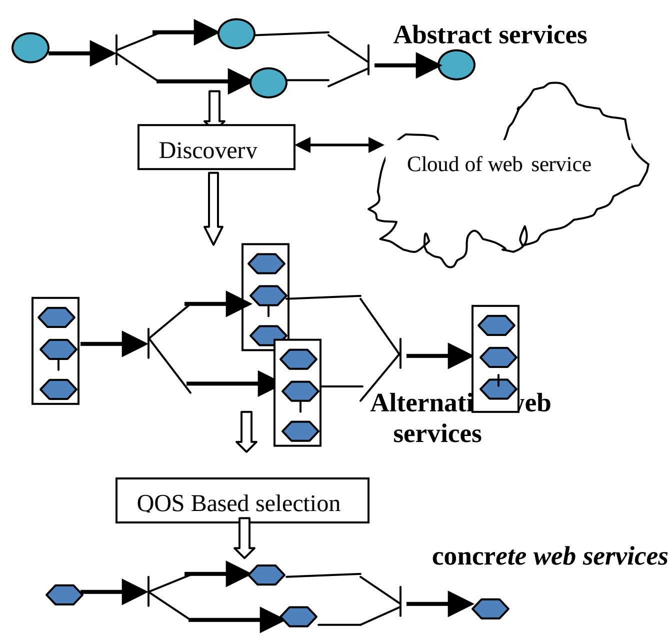
III Research Methodology / Material & Methods in Details

A.Abstract and Composite Web Services:

As shown in Fig.01 we distinguish in the composition process between the following two concepts:

- An abstract composite service, which can be defined as an abstract representation of a composition request CSabstract = {S1, . . . , Sn}. CSabstract refers to the required service classes (e.g. flight booking) without referring to any concrete web A concrete composite service, which can be defined as an instantiation of an abstract composite service. This can be obtained by binding each abstract service class in CSabstract to a concrete web service sj , such that sj € Sj . We use CS to denote a concrete composite service.

Input :Abstract Composition



output :Concrete Composition

Fig. 01 : Conceptual overview of Web Services

B. Web Service Composition:

Web Service compositions can be seen as workflows based on Web Services. As depicted in Figure02, there is a workflow model that consists of abstract tasks describing the required functionality (e.g. invoking a credit rating) of a specific workflow step. One of the main issues hereby is the selection of appropriate Web Services

ISSN : 2278 – 1021

International Journal of Advanced Research in Computer and Communication Engineering,
Vol. 1, Issue 7, September 2012

that form the execution plan for a Web Service composition. The functionality of each task can be provided by different candidate Web Services. Web Services that provide similar or identical functionality are grouped in the same category. Web Services within the same category may have different non-functional attributes.

Definition: A sequential Web Service composition consists of *n* tasks. Task *i* (*i*=1,...,*n*) will be executed before task *i'* (*i'*=1,...,n) if *i*<*i'*. The set of *mi* different candidate Web Services that provide the required functionality for task *i* is called category *i*. A binary variable *xi,j* is introduced. X*i,j*=1 means that Web Service *j* of category *i* is selected for being executed within the execution plan. To ensure that only one Web Service per task is selected, it is necessary that

$$\sum_{j=1}^{m_i} X_{i,j} = 1 \qquad \forall\, i=1,\ldots,n$$

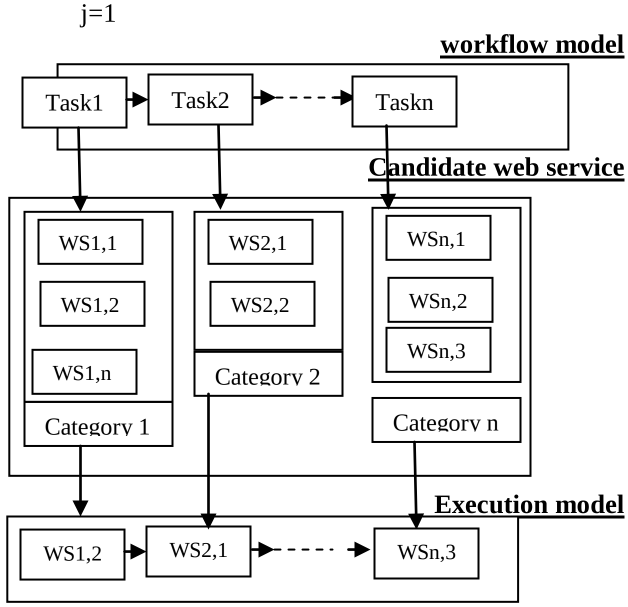


Fig. 02 : Web Service Composition

C. QoS Criteria:

In our study we consider quantitative non-functional properties of web services, which can be used to describe the quality criteria of a web service.

These can include generic QoS attributes like response time, availability, price, reputation, throughput etc, as well as domain-specific QoS attributes like bandwidth for multimedia web services as long as these attributes can be quantified and represented by real numbers. We use the vector Qs = {q1(s), . . . , qr(s)} to represent the QoS attributes of service 's', where the function qi(s) determines the value of the i-th quality attribute of 's'. The values of these QoS attributes can be either collected from service providers directly (e.g. price), recorded from previous execution monitoring (e.g. response time) or from user feedbacks (e.g. reputation). The set of QoS attributes can be divided into two subsets: positive and negative QoS attributes. The values of positive attributes need to be maximized (e.g. throughput and availability), whereas the values of negative attributes need to be minimized (e.g. price and response time). For the sake of simplicity, here we consider only negative attributes (positive attributes can be easily transformed into negative attributes by multiplying their values by -1).

Aggregation Type	Examples	Function
Summation	**Response Time**	$q'(CS) = \sum_{j=1}^{n} q(s_j)$
	Price	
	Reputation	$q'(CS) = 1/n \sum_{j=1}^{n} q(s_j)$
Multiplication	**Availability**	$q'(CS) = 1/n \prod_{j=1}^{n} q(s_j)$
	Reliability	
Minimum	**Throughput**	$q'(CS) = 1/n \min_{j=1}^{n} q(s_j)$

D.QoS Computation of Composite Services:

The QoS value of a composite service is decided by the QoS values of its component services as well as the composition model used (e.g. sequential, parallel, conditional and/or loops). In this paper, we focus on the sequential composition model. Other models may be reduced or transformed to the sequential model. Techniques for handling multiple execution paths and

Unfolding loops from, can be used for this purpose.

The QoS vector for a composite service CS is defined as:

Q_{CS} = {q'01(CS), . . . , q'0r(CS)}.

Q'i(CS) represents the estimated value of the i-th QoS attribute of CS and can

be aggregated from the expected QoS values of its component services.

In our model we consider three types of QoS aggregation functions:

1) summation, 2) multiplication and 3) minimum relation.

ISSN : 2278 – 1021

International Journal of Advanced Research in Computer and Communication Engineering,
Vol. 1, Issue 7, September 2012

Table 1 shows examples of these aggregation functions.

IV QoS Aware Service Composition and Analysis

A.Web Service QoS Model:
There are two layers AS and CS while composing web services. Fig. 03 describes the relationship between them.

Table 01 : Examples of QoS Aggregation Functions

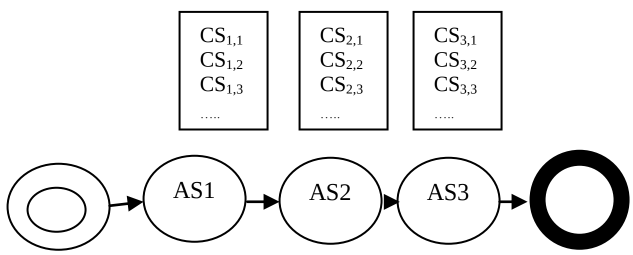


Figure 03. Relationship Between AS and CS

AS(*i*) stands for abstract web service *i*. *CS*(*i*, *j*) stands for the *jth* concrete candidate for *AS*(*i*). In short, our work is on how to choose suitable CS for each AS. The simplest situation is that there are only one AS and multiple candidate CSs for it. One CS should be chosen to meet QoS requirement.

TABLE 2 shows composite attributes for four basic workflow patterns.

Pattern	Sequence	Fork	Choice	Loop
Composite Cost (C)	$\sum_{i=1}^{n} c_i$	$\sum_{i=1}^{n} c_i$	$\sum_{i=1}^{n} p_i c_i$	n×c
Composite Latency (L)	$\sum_{i=1}^{n} l_i$	$\sum_{i=1}^{n} l_i$	$\sum_{i=1}^{n} p_i l_i$	n×l
Composite Availability (A)	$\prod_{i=1}^{n} a_i$	$\prod_{i=1}^{n} a_i$	$\sum_{i=1}^{n} p_i a_i$	a^n
Composite Reliability (R)	$\prod_{i=1}^{n} r_i$	$\prod_{i=1}^{n} r_i$	$\sum_{i=1}^{n} p_i r_i$	r^n
n : is consider as node number $\sum_{i=1}^{n} p_i = 1$: p_i : **is probability of chosen node**				

Table 02 : Attribute Pattern Table for Web Services

In this synopsis, four QoS attributes are considered, they are cost(*c*), latency(*l*), availability(*a*) and reliability(*r*). Single Web Service QoS can be represented as (*c*, *l*, *a*, *r*). In general, better quality always means higher availability and reliability, and lower cost and latency. Composite web service QoS is based on single web services. According to, four common composition patterns and four basic quality attributes are analysed here. For each pattern, quality value calculation for composite web services can be totally different.
A real web service composition will compose some of the above patterns to form a useful application.

B.General Web Service Composition Qos Model :
The web service composition model is constructed as follows:
There are *n* nodes in the process. And node *i* is represented as *AS*(*i*). For each *AS*(*i*), there are *ti* CSs for it and each is called *CS*(*i*, *j*). For each *CS*(*i*, *j*), the quality is considered as a vector < *cij*, *lij*, *aij*, *rij* >, where *cij* means cost, *lij* means latency, *aij* means availability and *rij* means reliability. So the quality for each *AS*(*i*) is represented as following matrix.

$$AS(i) = \begin{bmatrix} Ci1 & Ci2 \ldots\ldots\ldots\ldots Cit_i \\ Li1 & Li2 \ldots\ldots\ldots\ldots Lit_i \\ Ai1 & Ai2 \ldots\ldots\ldots\ldots Ait_i \\ Ri1 & Ri2 \ldots\ldots\ldots\ldots Rit_i \end{bmatrix} \quad (0<i\leq 1)$$

And the best quality web services for composition meet the following constraints (Ki means the chosen number of Cs for node i)

$C(C_{1,k(1)}, C_{2,k(2)} \ldots\ldots C_{n,k(n)}) \leq$ maxcost
$L(L_{1,k(1)}, L_{2,k(2)} \ldots\ldots L_{n,k(n)}) \leq$ maxlatency
$A(A_{1,k(1)}, A_{2,k(2)} \ldots\ldots A_{n,k(n)}) \leq$ maxavailabi
$R(R_{1,k(1)}, R_{2,k(2)} \ldots\ldots R_{n,k(n)}) \leq$ maxreliabili

And the objective function is :

$$\mathbf{Min\ F(CS_{1,k(1)}, CS_{2,k(2)} \ldots\ldots CS_{n,k(n)})}$$

The first four constraints are global constraints, and the last is objective function. The objective function can be represented as different styles based on different user

ISSN : 2278 – 1021

International Journal of Advanced Research in Computer and Communication Engineering,
Vol. 1, Issue 7, September 2012

requirements. Assuming availability and reliability are always above some value, only cost and latency are discussed in this proposal. And it's not difficult to extend to include availability and reliability if necessary.

V.Enabling Web Services Composition with Software Agents:

So, We proposed a framework to support negotiation during QoS aware Web Service Composition. We add two layers between service requirement and web service candidates, as shown in figure 04. In the first layer, each Web Service candidate is linked to a home norm base, which can be used by home agent to negotiate among other home agents. Norms are a set of rules and regulations, an under-lying protocol governing the agent communications network and agents complex behaviour. Norms revolves around agents, which influences the agents to execute a series of concerted actions to achieve a particular goal. And the second layer is composition algorithm which links with the home agent for choosing the best services for the composition.

Definition 2 Concrete Execution Path (CEP): A concrete execution path is a path in which each node is a concrete (real) service that corresponds to an abstract service on an abstract execution path. An abstract execution path can have more than one concrete execution path.

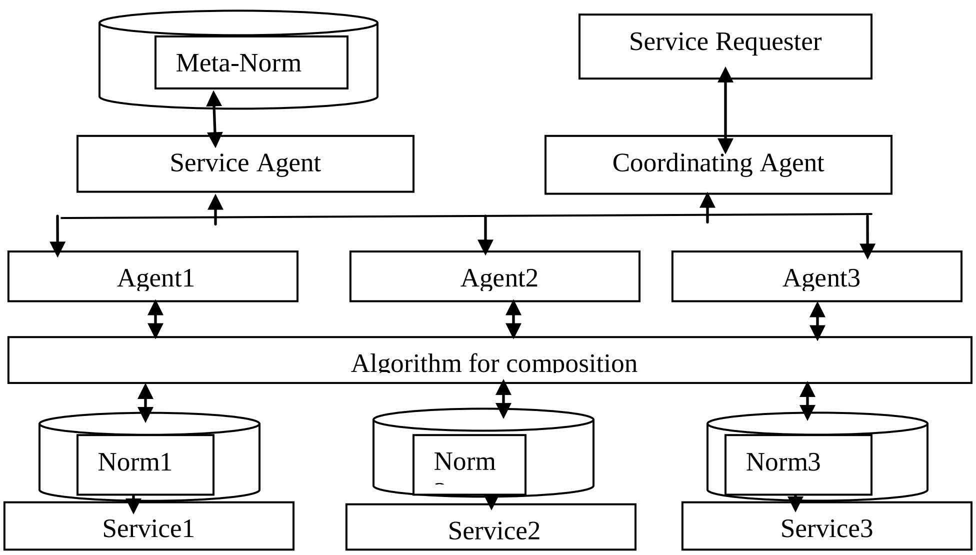


Fig 04 : Proposed Framework Using Agents

VI.QoS-Based Web Service Composition with MO ACO :

As a basic control strategy, *divide and conquer* is widely used to solve complex problems. This strategy is adopted in our study by decomposing composite services with a general flow structure into parallel execution paths, each of which is essentially a sequential flow structure. The maximum number of parallel execution paths is determined by the number of AND split structure patterns in the composite service.

Definition 1 Abstract Execution Path (AEP): An abstract execution path is defined as a sequential path from the starting point to ending point in the functional graph, which includes only one branch in conditional operations (starting at an XOR split) and one branch in parallel operations (starting at an AND split), and does not include any abstract services that have been included in any other abstract path.

Definition 2 Concrete Execution Path (CEP): A concrete execution path is a path in which each node is a concrete (real) service that corresponds to an abstract service on an abstract execution path. An abstract execution path can have more than one concrete execution path.

ISSN : 2278 – 1021

International Journal of Advanced Research in Computer and Communication Engineering,
Vol. 1, Issue 7, September 2012

Definition 3 (General MOP): A general MOP is to find a vector $\vec{x}$ = [x1, x2, x3, . . . , xn] which optimizes the vector function

$$f(\vec{x}) = [\, f1(\vec{x}), f2(\vec{x})\ldots\ldots fm(\vec{x})]$$

where xi (i = 1, . . . , n) are called decision variables (parameters) & fi($\vec{x}$), i = 1, . . . , m, are the objective functions for the corresponding m objectives.

1. Rule for Minimization Attributes:

$$cs.a^i = \begin{cases} \dfrac{a^i_{\ max} - cs.a^i}{a^i_{\ max} - cs.a^i} & \text{if } a^i_{\ max} \neq a^i_{\ min} \\ 1 & \text{If } a^i_{\ max} = a^i_{\ min} \end{cases}$$

2. Rule for Maximization Attributes:

$$cs.a^i = \begin{cases} \dfrac{cs.a^i - a^i_{\ min}}{a^i_{\ max} - a^i_{\ min}} & \text{if } a^i_{\ max} \neq a^i_{\ min} \\ 1 & \text{If } a^i_{\ max} = a^i_{\ min} \end{cases}$$

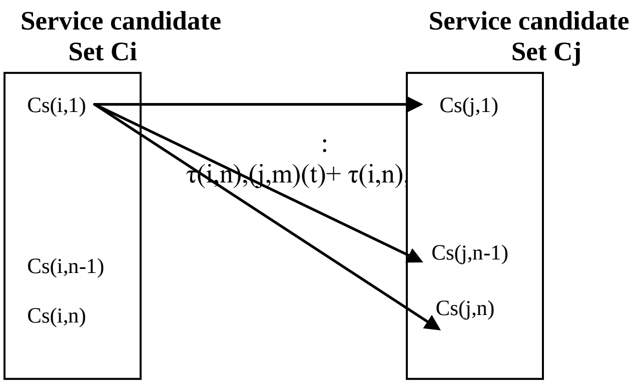


Fig 05: Pheromone on Edges

The pheromone is updated according to the formula:

$$\tau(i,n),(j,m)= \tau(i,n),(j,m)(t)+ \Delta\tau(i,n),(j,m)$$

where $\Delta\tau(i,n),(j,m)$ is the quantity of pheromone laid on the edge between web service cs(i,n) and web service cs(j,m) which is intuitively given as below:

$\Delta\tau(i,n),(j,m) = [\Delta\tau^1(i,n),(j,m)\ldots.\Delta\tau^K(i,n),(j,m)]$ <u>if edge is visited</u>

$\Delta\tau(i,n),(j,m) = 0$ <u>if no edge visited</u>

Where :

$$\Delta\tau(i,n),(j,m) = \frac{Gw/Lp(f(\vec{x}))}{|\, f^0 - f_w(\vec{x})|} \quad (w \in \{1..k\})$$

And G1 to Gk are constant set according to the weight of each attributes. Particularly, the pheromone updating rule as

$$\frac{Gw/Lp(f(\vec{x}))}{|\, f^0 - f_w(\vec{x})|} \quad (w \in \{1..k\})$$

Instead as:

$$\frac{Gw}{|\, f^0 - f_w(\vec{x})|} \quad (w \in \{1..k\})$$

The reason for this is based on the consideration that we prefer those solutions that can approach ideal value of each objective in the ideal vector as close as possible and meanwhile, minimize the multi objective utility function or else, the solutions that can perfectly optimize some objective but may badly fail to optimize the other objectives can be reinforced.

VII.Conclusion

We propose a strategy to decompose a composite services with a general flow structure into parallel execution paths. We then model dynamic service composition for each execution path as a multi-objective optimization problem, and present a new version of ACO algorithm, MO ACO with the agent to store web service specification, to handle this problem. The study illustrate that our MO ACO approach can find near-optimal solutions on the fly for multi-objective problems out of a huge search space in a very efficient way. The literature survey also reveal that MO ACO is scalable to support composition of very complex web services and agent based technology utilize minimum bandwidth and memory. In the research field of ACO, some

ISSN : 2278 – 1021

International Journal of Advanced Research in Computer and Communication Engineering,
Vol. 1, Issue 7, September 2012

improvement has been proposed to further improve the performance of ACO algorithms. We will explore the possibility to integrate the improvement strategy into our MO ACO with agent based technology in order for achieving even better QoS of composite service from dynamic web service composition by ant colony optimization with agents as web service.

References

[1] Rainer berbner, Ralf Steinmetz "Heuristics of QoS-aware Web service Composition" IEEE International Conference on Web Services (ICWS'2006)

[2] Mohammad Alrifai, Thomass Risse "combining Global optimization with local selection for efficient QOS-Aware Service composition" ICW3C2 , april 20-24,2009, Madrid Spain , ACM 978-1-60558

[3] Tao ZHOU, Deren Chen and Xiaolin ZHENG " Objective function analysis for QoS-Aware web service Composition" IEEE Internationa conference on e-Bussiness Engineering (ICEBE'06)-7695-2645-4/06 IEEE

[4]Mihhail Matskin, sobah abbas Peterson " Enabling Web Service Composition with software agents" Norweign University, misha@idi.ntnu.no

[5]G Vadivelou, E.IIvarasan, S. Prasanna " algorithm for web service composition using Multi-Agents" International Journal of computer Application (0975-8887) Volume 13- No.8, January 2011

[6]Tao Yu, Yue Zhang " Efficient Algorithms for Web services Selection with end to end QoS constraints" ACM transaction on web, vol 1, No. 1, Article 6, May 2007

[7] M. adel serhani, rachida Dssouli " A QoS broker based architecture for efficient web service selection " proceeding of IEEE international conference on web services ICWS'05-7695-2409-5/05

[8] Florian rosenberge, Max Benjamine muller " Metaheuristic Optimization of large scale QoS – aware service Compositions" 2010 IEEE international conference on service computing, 978-0-7695-2010 IEEE

[9] Ekachai Jinhirunkul, peraphon sophatsathit " Web navigation analysis and simulation using ant Colony Optimization" Proceeding of the 4th international conference on autonomous Robot and agents Feb 2009-978-4244-2009-IEEE

[10] Amit Padovitz, Shonali Khrishnswamy and Seng Wai Loke " Towards efficient selection of web services" 1st International conference on Web services and agent based Engineering, Sydeney, Australia, July 2003

[11]Kristina Lerman "design and mathematical analysis agent based system" Information Science Institute, University of southern California, USA

[12]Wei Zhang, Carl Chang "QoS based dynamic Web service Composition with ant colony optimization" 34th annual IEEE computer software and application conference-0730-3157/10-2010 IEEE

[13]M. Erdem Tijrsem, M hadi Gunes " performance analysis of mobile agents using simulation" Bogazici University, Istanbul, Turkey

[14]Aki Vainio and Kimmo Salmenjko " Improving study planning with an agent based system" Faculty of Technology, university of Vaasa, Finland

[15] Wei-neng chen, Jun Zang "workflow scheduling in grids : an Ant colony Optimization approach " 4244-1340-2007 IEEE

[16]M A R Dantas and T G Ramos " An enhanced scheduling approach in a distributed parallel environment using mobile agents" proceeding of the 16th Annual symposium on high peroformance computing systems and applications (HPCS'02) 0-7695-1626@2002

[17] Giulianna Marega Marques, Ricardo Jose "performance analysis of scheduling techniques in distributed parallel environment" 978-1-4244-1706-3/08@2005 IEEE

[18] Fuyuki Ishikawa1, Nobukazu Yoshioka2, Yasuyuki Tahara2, Shinichi Honiden2 "Mobile Agent System for Web Services Integration in ervasive Networks" 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430, Japan, fishikawa,nobukazu,tahara,honideng@nii.ac.jp

[19] Viet-Ha Nguyen[1], The Duy Bui[1], Quang-Dung Vu[1], Nguyen Thi Thu Hien "Towards scalable agent based web service systems" 144 Xuan Thuy Str., Cau Giay Distr., Hanoi, Vietnam Email: hien ntt@vnu.edu.vn

[20] Abdelkader Outtagarts "Mobile Agent-based Applications : a Survey "IJCSNS International Journal of Computer Science and Network Security, VOL.9 No.11, November 2009

[21] Viet-Ha Nguyen, The Duy Bui, Quang-Dung Vu, Nguyen Thi Thu Hien "Towards scalable agent based web service systems "144 Xuan Thuy Str., Cau Giay Distr., Hanoi, Vietnam ,Email: hien ntt@vnu.edu.vn

[22] David Kotz and Robert S. Gray "Mobile Agents and the Future of the Internet" Dartmouth College Hanover, New Hampshire 03755 dfk@cs.dartmouth.edu,

[24] Toktam Ghafarian, Mohsen Kahani " semantic Web Services composition based on ant colony optimization method" 978-1-4244-2009 IEEE

[25] Vector Munoz, felix " Ant colony optimization for data grid replication services" Technical report RR-06-08. DIIS UNIZAR – april 2008

[26]Qi Zhao, yanliao Tan " A Load balancing based model for Dynamic web service selection" 2nd International symposium on computational intelligence and design, 978-0-7695- 2009 IEEE

ISSN : 2278 – 1021

International Journal of Advanced Research in Computer and Communication Engineering,
Vol. 1, Issue 7, September 2012

[27]Anbazhagan Mani, arun nagrajan " Understanding quality of services for web services" Report of IBM Software lab, India – manbazha @ in.ibm.com
[28] Viet-Ha Nguyen, the Duy Bui " towards scalable agent based web service systems" university of engineering and technology, hanv@vnu.edu.vn
[29] Hoops " Web services as distributed systems- part 2" posted in design and technologies , December 8,2010
[30]Camelia Chira, M pintea "New metaheuristics for distributed computing" Computational intelligence report (CIR) No. 062006, June 2006, ISSN 1841-995X
[31]Amir Padovitz, Shonali krishnaswamy " Towards efficient selection of web services" Monash University, padovitz@bigpond.com
[32]youngkon Lee " A Research for web service quality presentation methodology for SOA Framework" 6th International conference on advanced language processing and web information technology, 7695-2930-2007 IEEE
[33]Eludiora, Safiriyu "Load Balancing policy for distributed web service " International Journal of communication, network and system sciences- august 2010
[34]Emilio Mancini, Umberto villano " Mobile Agents self-optimization with autonomic web services" Springer verlag berlin Heidelberg – 2007
[35]Ruimini Liu, Feng Chen " Agent based web services evolution for pervasive computing" Proceeding of 11th Asia pacific software engineering (APSEC'04) 1530-1362/ 04 IEEE [36]Bin wu, Chaozhen Guo " The research and improvement on co-ordinated negotiation architecture of web service composition based on agent in condition of multi negotiation concurrency" proceeding of 14th International conference on computer supported cooperative work in design -978-1-4244/10- 2010 IEEE
[37] Gerardo Canfora, maria Vilani " an approach for QoS Aware service composition based on genetic algorithm" GECCO'05 ACM June 2005
[38] Julian day " selecting the best web service" university of Saskatchewan, SK, Canada, Julian,day@usask,ca
[39] Eyhab Al-Masri, Quasy mohmad " QoS Based discovery and ranking of web services" 1-4244-1251-2007 IEEE
[40] Zakaria maamar, Hamdi Yahyaoui " Towards an agent based and context oriented approach for web services composition" IEEE transaction on knowledge and data engineering, Vol. 17, no. 5, may 2005.

Biography

Mr. Sunil R Dhore is currently working as associate professor at Army Institute of Technology, Pune(MS) India. He Completed his ME(Computer Engg) from College of engineering Pune, currently pursuing his PhD from Amaravati University. His research area is Distributed System.

Dr.M U Kharat is working as Professor at MET's College of Engineering, Bujbal Knowledge City, Nashik(MS) India. His area of interest if High performance network and Distributed System