Web Service Composition Methods: A Survey

E. Pejman, Y. Rastegari, P. Majlesi Esfahani, A. Salajegheh

***Abstract*— Web service composition is one of the most challenging problems of recent years. The number of service providers is increasing, and along with that for a request they offer multiple services with the same functionality, so it makes the problem of composition quite complex. In this paper we present several methods on service composition. We group them into two categories: methods based on evolutionary algorithms (e.g. Genetic Algorithm) and methods based on non-evolutionary algorithms (e.g. Dynamic Programming, Heuristic Algorithms). In this paper we focus on syntactic service compositions based on Quality of Service (QoS).**

***Keywords*- Web service; Quality of service; Service composition; Evolutionary algorithm.**

I. INTRODUCTION

Web service is a promising technology that allows constructing and sharing independent and autonomous software [26]. Service interface publication, service discovery and service invocation are performed using XML-based standards, known as WSDL, UDDI and SOAP [15].

The Web service model consists of three entities, the service provider, the service registry and the service consumer. Fig. 1 shows a graphical representation of the traditional Web service model [24].

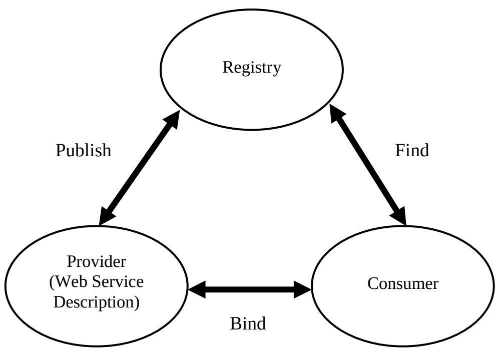

Figure 1. Web service model

Manuscript submitted in January 18, 2012. Accepted in January 27,2012, Revised February 7, 2012.

Emad Pejman is with the Islamic Azad University-South Tehran Branch, Department of Computer Science and Software Engineering, Tehran, Iran. (e-mail: emad.pejman@gmail.com).
Yousef Rastegari is with the Shahid Beheshti University, Department of Electrical Engineering and Computer Science, Tehran, Iran. (e-mail: y_rastegari@sbu.ac.ir com).
Pegah Majlesi Esfahani is with the Islamic Azad University-North Tehran Branch, Department of Computer Engineering, Tehran, Iran. (e-mail: pegahmajlesi@gmail.com).
Afshin Salajegheh is with the Islamic Azad University-South Tehran Branch, Department of Computer Science and Software Engineering, Tehran, Iran. (e-mail: salajegheh@iau.ac.ir).

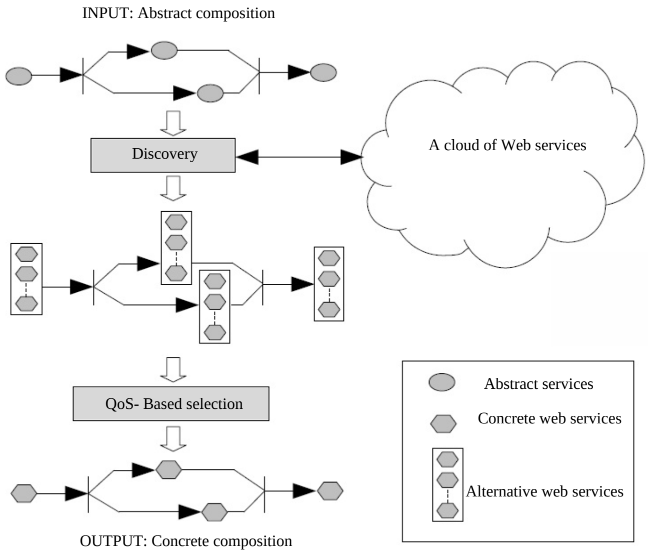

Figure 2. Service composition model

By adopting standard-based protocols SOAP, WSDL and UDDI, service components from different service providers can be conveniently integrated into a composite service regardless of their locations, platforms and/or execution speeds.

With growing demands, many service providers have started to offer different QoS service levels, to meet the need of different user groups [1]. Web service composition has received increasing attention by the research community in the past few years [25].

Fig. 2 gives a conceptual overview of the QoS-aware service composition problem. Given an abstract composition request, which can be stated in a workflow-like language (e.g. BPEL), the discovery engine uses existing infrastructure (e.g. UDDI) to locate available web services for each task in the workflow using syntactic (and probably semantic) functional matching between the tasks and service descriptions. As a result, a list of candidate web services is obtained for each task with different QoS attributes. The goal of QoS composition is to select one candidate service from each list such that the aggregated QoS values satisfy the user's end-to-end QoS requirements (constraints). In service oriented environments, where deviations from the QoS estimates occur and decisions upon replacing some services has to be taken at run-time, the efficiency of the applied selection mechanism becomes crucial [22].

In this paper we discuss methods shown in Fig. 3. We group these methods in two categories: methods based on evolutionary algorithms (shown in dashed outline) such as Genetic Algorithm, Immune System Algorithm, PSO, etc. and methods based on non-evolutionary algorithms such as Dynamic Programming, Graph Theory, Integer Programming, etc. In this paper we discuss advantages and disadvantages of each group.

ISBN: 978-988-19251-1-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

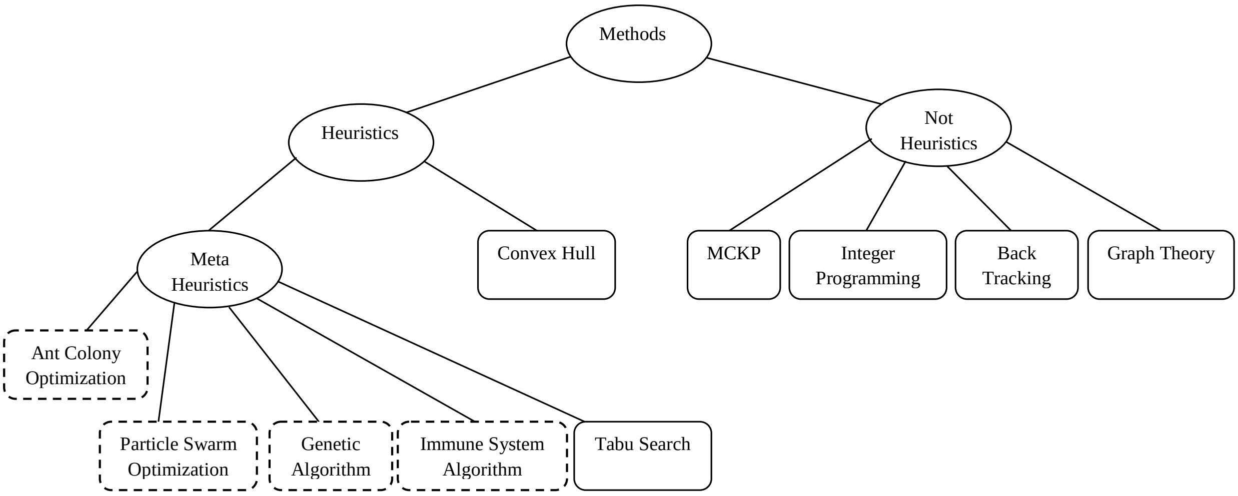

Figure 3. Methods tree chart
Dashed outline: Methods based on evolutionary algorithms
Simple outline: Methods based on non-evolutionary algorithms

The reminder of this paper is organized as follows: In section II we discuss methods based on non-evolutionary algorithms. Section III we discuss methods based on evolutionary algorithms, and finally we conclude the paper with comparing these categories in section IV.

II. METHODS BASED ON NON EVOLUTIONARY ALGORITHMS

In this part we discuss several methods for service composition based on QoS attributes. These methods are based on non-evolutionary algorithms.

Yu, T. et al. propose several algorithms on service selection using QoS broker (A QoS broker collects QoS information about servers and makes decision (service candidates) for clients). In [1, 2] they propose two methods for service selection problem. The goal is to maximize the utility function while satisfying the constraints. Constraints are clients QoS requirements that should be satisfied in an end to end service composition (e.g. total Cost should not be more than 300$). There are several forms of to model a business process, like sequential, parallel, loop, etc. They propose their algorithm on sequential form and use only one constraint for composition problem. They use several QoS attributes like, Response time, Cost, Reliability and Availability.

They represent two different approaches for solving the problem. The first method for selection is called Combinatorial approach. They model the problem as Multiple-Choice Knapsack Problem (MCKP) that each item has a weight and a profit, and knapsack has a capacity. Each item in each service class is a candidate service that profit and weight presents service utility and QoS attribute respectively. Knapsack capacity is global constraint. The algorithm is to select an item from each service class such that the total utility under constraint is maximized. They formulate the problem as shown (1).

$$\begin{array}{ll} Max & \sum_{i=1}^{k}\sum_{j\in s_i} F_{ij}\,x_{ij} \\ Subject\ to & \sum_{i=1}^{k}\sum_{j\in s_i} R_{ij}\,x_{ij} \le R \\ & ,\sum x_{ij} = 1 \end{array} \tag{1}$$

F_{ij}: *Utility value at step i for candidate j*
r_{ij}: *Response time of candidate j at step i*
R: *Total response time*

They use Dynamic programming to solve the problem and with using Pisinger's algorithm they efficiently find a solution.

The second approach is graph theory. They model service composition to a constrained shortest path problem. In this approach each service candidate in each service class represents a node. They move QoS parameter from nodes to their corresponding edges and construct a DAG. Then by using several algorithms like Constrained Bellman-Ford (CBF), Constrained Shortest Path (CSP) they solve the problem.

In [3] Yu, T. et al. consider Multiple QoS Constraints. They use different form of workflow in their business process and with using an efficient heuristic algorithm they tackle the problem [4]. According to their algorithm, Yu, T. et al. design a framework for service composition [18].

Chen, H. et al. designs the QoS-Capable Web Service architecture QCWS [5] that implements a QoS broker between Web service clients and providers.

Zeng et al. [6] present a platform named AgFlow that enables the quality-driven composition of Web services. In this platform QoS of web services is evaluated and then it selects web services that can optimize total QoS, it also takes into account users requirements. They consider several QoS attributes such as execution price, execution duration, reputation, successful execution rate and availability. They also consider several forms of workflow such as AND/OR and loop. They propose two approaches for service selection in the composition, local optimization and global planning. In local optimization approach they use SAW [7] technique to compute the overall QoS of each web service in order to

ISBN: 978-988-19251-1-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

select the optimal one, and then with formula (2) they compute the Score of web services.

$$Score(S_i) = \sum_{j=1}^{5}(V_{ij} * W_j) \quad (2)$$

V_{ij} : Matrix of QoS Attributes for services
W_j : User define weight for each QoS attribute
i: Represent number of tasks
j: Represent number of QoS attributes

In local optimization approach, service selection is performed individually for each task and a service with best score is chosen. In global planning they take into account the global constraints. They use IP (Integer Programming) [8] to solve the problem. Although it is better than exhaustive searching but running time in IP is considerable. By unlooping loops in workflow, they also solve this form of workflows in the business process.

In [9] they propose a method for service selection and composition in runtime. They present backward context based service selection named (BCCbSS). The main idea is that the algorithm goes through the process and selects services step by step. After selecting services at each step, the algorithm goes back and checks if the selected services from previous step are best for composition or not and then it invokes them. Their algorithm only works for sequential model and it cannot find the optimal composition. But this method is a fault tolerant one, it means in runtime if selected services wouldn`t working properly it selects next best services for that task.

R. Berbner et al. design an architecture WSQoSX (Web Services Quality of Service Architectural Extension) [11, 12], that supports late binding of Web Services at runtime as well as dedicated accounting and monitoring mechanisms [10]. In [10] they propose heuristics for solving composition problem. They use an aggregation function to compute the utility of the overall service composite. They present their algorithm on the sequential form and consider several QoS attributes. As they mentioned, the LP relaxation of the MIP formulation of composition problem is solved using a standard algorithm (e.g. simplex), then based on the result of relaxed integer program, a backtracking algorithm is used to find a solution. Constraints checked as the algorithm is running. They also present some meta-heuristics for the improving of feasible solution.

Also [13] presents a heuristic algorithm to solve the composition problem. Similar to other method they use several QoS criteria. At first they formulate the problem to a Multi-Choice Knapsack problem and consider only one global constraint. By applying Multiple Criteria Decision Making (MCDM) theory [7] they merge multiple resources in to a whole. For local selection they use greedy algorithm and select web services with highest score, for global planning they use the idea of convex hull [14]. They assume each candidate represents a point, that x-coordinate represents constraint and y-coordinate represents QoS score. So each task is a set of points. For each task (set of point) they construct a convex hull using Graham-scan or Quick-hull algorithm. In order to find optimal solution they sort the frontiers in descending order and select segments with greater gradient while meeting constraint. They can also consider multiple constraints in their method. In [20] they also try to tackle the problem with the idea of convex hull.

Alrifai et al. [21] propose an algorithm based on heuristics that decompose the main problem into sub problems and with solving these, according to their experiments, they found near to optimal solution. They use aggregation functions for computing overall QoS attributes. They take into account global constraints and because of the decomposition of QoS problem, they also decompose global constraints into n local constraints and these local constrains are checked in each of local selections. Evaluation shows that the algorithm performance is better than IP programming [6] and heuristic algorithm WSHEU [4]. In [22] they use the same approach except beside using heuristics, they use mixed integer programming (MIP) to find the optimal composition.

So far we discuss several methods and algorithms in service composition. Approaches that use Integer Programming, Dynamic Programming or Graph Theory, definitely find best solution, But complexity in these algorithms is high. Although, as stated earlier, some heuristic algorithms helps these algorithms to converge sooner.

III. METHODS BASED ON EVOLUTIONARY ALGORITHMS

In this part we discuss methods based on evolutionary algorithms.

In paper [15] they use Genetic algorithm to tackle the composition problem. They use fitness function to compare solutions. Constraints are considered in fitness function and all forms of workflow in business process are taken into account. The approach for computing the overall QoS in composition is similar to Cardoso [16] except for loop that they use the method in [6].

First they encode the problem in to a genome and then Genetic operation performed on that. Fitness function maximizes some QoS attributes while minimizing some of them and also a penalty for meeting constraints is also considered. The fitness function they use is shown in formula 3.

$$F(g)=\frac{w_1\,Cost(g) + w_2\,Response\ Time(g)}{w_3\,Availability(g) + w_4\,Reliability(g)} + w_5\,D(g) \quad (3)$$

W_i : User define weight for each QoS attribute
$D(g)$: a penalty for meeting constraints

Jiuyun et al. [17] use immune algorithm to tackle the composition problem. First they encode the problem in to an antibody. This method has two steps: the Immune selection operation and Clonal selection operation. In immune selection, antibodies is proliferated and suppressed in order to control their density in the mating pool and also make sure antibodies that are helpful and potential (vaccine) will not be destroyed. In Clonal operation they use antibodies with high fitness as heuristic information for speeding up convergence. They consider several control flow operators of business process and take into account several QoS

ISBN: 978-988-19251-1-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

attributes in fitness function like service cost, service response time, service availability and service reliability. Antibodies with best fitness are considered as vaccine in the algorithm.

[19] proposes an algorithm named ACAGA_WSC. They use the combination of ant colony and Genetic algorithm for solving service composition problem. First they model service composition in to Ant Colony algorithm. but, as they mentioned, there are some important parameters in Ant Colony algorithm that have great effect on the algorithm and ones when we select them trough experiments, those are not suitable for other experiments, so they use Genetic algorithm to set the key parameters of the Ant Colony algorithm in order to obtain the great efficiency.

In [23] authors solve the problem with combination of Genetic and Tabu search algorithm. They compare their method only with genetic algorithm.

Chen, M. et al. [27] solve the problem of composition by using DPSO (Discrete Particle Swarm Optimization). In this method they consider each particle as a solution. Each particle has a position and a velocity. Particles try to change their positions according to two elements: first their last best position and second the best position that has been seen so far. They change their position and velocity by using formula 4.

$$V = C_1 (X_{Pbest} - X) + C_2 (X_{Gbest} - X) \qquad (4)$$
$$X = X + V$$

They consider different forms of workflow. They use formula (2) to compute the score.

IV. CONCLUSION

Interaction between services transform the world of information oriented internet to a cloud of services. In this wide and distributed environment we need service composition to answer different requests. Service composition is a NP-hard problem. The problem is by getting a composite plan and service candidates for each task in workflow languages (e.g. BPEL) as inputs, we need to find the best solution for the composition.

In this paper we studied several methods in service composition. We grouped them in two categories: methods based on evolutionary algorithms and methods based on non-evolutionary algorithms. Methods in first category definitely find an optimal solution to the problem but, complexity in these methods is high. Although, some heuristics help them to speed up and improve the complexity. Methods in second category converge faster but the main problem in this category is that the final solution is not necessarily optimal.

So we suggest for small scale environment or locations that business processes are simple or candidates are limited, using non-evolutionary algorithms is more efficient and in environments that the business processes are complex and service candidates are distributed, using methods based on evolutionary algorithms can achieve best solution in a reasonable time.

REFERENCES

[1] T. Yu, K.J. Lin, “Service Selection Algorithms for Web Services with End-to-end QoS Constraints,” Journal of Information Systems and E-Business Management, Vol. 3, Number 2, Springer 2005.
[2] T. Yu, K.J. Lin, “Service Selection Algorithms for Web Services with End-to-end QoS Constraints,” e-Commerce Technology CEC 2004, Proceedings. IEEE International, pp. 129 – 136, August 2004.
[3] T. Yu, K.J. Lin, “Service Selection Algorithms for Composing Complex Services with Multiple QoS Constraints,” In Proceedings of ICSOC, pp. 130-143, 2005.
[4] T. Yu, Y. Zhang, K.J. Lin, “Efficient algorithms for Web services selection with end-to-end QoS constraints”, Journal ACM Transactions on the Web (TWEB), Volume 1, Issue 1, May 2007.
[5] H. Chen, T. Yu, K.J. Lin, “QCWS: An Implementation of QoS Capable Multimedia Web Services,” Proc. of IEEE 5th International Multimedia Software Engineering, Taiwan, December 2003.
[6] L. Zeng, B. Benatallah, M. Dumas, J. Kalagnanam, and Q.Z. Sheng, “Quality Driven Web Service Composition,” Proceeding of 12th International World Wide Web Conference (WWW), 2003.
[7] H.C. –L, K. Yoon, “Multiple Criteria Decision Making,” Lecture Notes in Economics and Mathematical Systems, , Springer 1981.
[8] H. Karloff, “Linear Programming,” Birkhauser, 1991.
[9] H.Q. Yu, and S. Reiff-Marganiec, “A Backwards Composition Context Based Service Selection Approach for Service Composition,” Services Computing SCC '09, IEEE International, pp. 419 – 426, October 2009.
[10] R. Berbner, M. Spahn, N. Repp, O. Heckmann, R. Steinmetz, “Heuristics for QoS-aware Web Service Composition,” International, pp. 72 – 82, December 2006.
[11] R. Berbner, T. Grollius, N. Repp, O. Heckmann, E. Ortner, and R. Steinmetz, "An approach for the Management of Service-oriented Architecture (SoA) based Application Systems," Enterprise Modelling and Information Systems Architectures (EMISA 2005), Klagenfurt, Austria 2005.
[12] R. Berbner, O. Heckmann, and R. Steinmetz, "An Architecture for a QoS driven composition of Web Service based Workflows," Networking and Electronic Commerce Research Conference (NAEC 2005), Riva Del Garda, Italy, 2005.
[13] D. Liu, Z. Shao, C. Yu, G. Fan, “A Heuristic QoS-Aware Service Selection Approach to Web Service Composition,” Proceeding ICIS '09 Proceedings of the 2009 eight IEEE, ACIS International Conference on Computer and Information Science, pp. 1184-1189.
[14] M. M. Akbar, M. S. Rahman, M. Kaykobad, “Solving the multidimensional multiple-choice knapsack problem by constructing convex hulls,” Computers and Operations Research, pp. 1259-1273, 2006.
[15] G. Canfora, M. Di Penta, R. Esposito, M. Luisa Villani, “An Approach for QoS-aware Service Composition based on Genetic Algorithms,” Proceeding GECCO '05 Proceedings of the 2005 conference on Genetic and evolutionary computation ACM, New York, 2005.
[16] I. Grossmann, “Review of nonlinear mixed-integer and disjunctive programming techniques,” Optimization and Engineering, pp. 227–252, September 2002.
[17] J. Xu, S. Reiff-Marganiec, “Towards Heuristic Web Services Composition Using Immune Algorithm,” Web Services ICWS '08, IEEE International, pp. 238 – 245, November 2008.
[18] T. Yu, K.J. Lin, “A Broker-Based Framework for QoS-Aware Web Service Composition,” e-Technology, e-Commerce and e-Service, Proceedings EEE '05. , pp. 22 – 29, April 2005.

ISBN: 978-988-19251-1-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

[19] Z. yang, C. shang, Q. liu, C. zhao, “A Dynamic Web Services Composition Algorithm Based on the Combination of Ant Colony Algorithm and Genetic Algorithm,” Journal of Computational Information Systems, pp. 2617-2622, 2010.

[20] V. Vanitha, V. Palanisamy, “A Global QoS-Aware Service Composition in Wireless Sensor Networks,” International Journal of Computer Applications, Vol. 1, No. 19, pp. 0975 – 8887, 2010.

[21] M. Alrifai, T. Risse, P. Dolog, and W. Nejdl, “A Scalable Approach for QoS-based Web Service Selection,” Book Service-Oriented Computing, ICSOC 2008 Workshops Springer-Verlag, Berlin, Heidelberg, 2009.

[22] M. Alrifai, T. Risse, “Combining Global Optimization with Local Selection for Efficient QoS-aware Service Composition,” Proceedings of the 18th international conference on World wide web, ACM, 2009.

[23] S. Bahadori, S. Kafi, K. Zamani far, M. R. Khayyambashi, “Optical Web Service Composition Using Hybird GA-Tabu Search,” Journal of Theoretical and Applied Information Technology, 2009.

[24] S. Dustdar, W. Schreiner, “A survey on web services composition,” Journal International Journal of Web and Grid Services ,Vol. 1, Issue 1, August 2005.

[25] V. Cardellini, E. Casalicchio, V. Grassi,; F. Lo Presti, “Flow-Based Service Selection for Web Service Composition Supporting Multiple QoS Classes,” Web Services ICWS 2007, IEEE , pp. 743 – 750, July 2007.

[26] J. El Haddad, M. Manouvrier, G. Ramirez, M. Rukoz, “QoS-driven Selection of Web Services for Transactional Composition, “International Web Services ICWS '08, IEEE, pp. 653 – 660, November 2008.

[27] C. Ming, W. Zhen-wu, "An Approach for Web Services Composition Based on QoS and Discrete Particle Swarm Optimization," Eighth ACIS International Conference on Software Engineering, IEEE, 2007.