Heuristics and Non-Heuristics Based Service Composition Methods

Aditya Khamparia¹ and Babita Pandey²

¹School of Computer Science and Engineering, Lovely Professional University, Phagwara, Punjab, India-144802, email: khamparia.aditya@gmail.com

²School of Computer Applications, Lovely Professional University, Phagwara, Punjab, India – 144802, email: shukla_babita@yahoo.co.in

Abstract:

Web service composition is one of challenging problem among research and industry groups. Lots of human efforts are required to extract useful information out of located information. There is need to composite web services semantically as number of service providers and consumers are increasing at an alarming rate. This paper is based on comparative study of various evolutionary approaches (e.g. Genetic algorithms, PSO) and non-evolutionary approaches (e.g. linear programming, graph theory). Various heuristic approaches are discussed based on quality of service attributes to facilitate easier composition.

Keywords: Genetics, PSO, Meta heuristics, web service, Knapsack.

1. Introduction

A Web service is a software interface that describes a collection of operations that can be accessed over the network through standardized XML messaging. It uses protocols based on the XML language to describe an operation to execute or data to exchange with another Web service. Web services operation can be carried out by XML based standards such as UDDI for locating the service or service discovery, detailed specification can be acquired from WSDL, client-server request and response handling can be managed by SOAP. Web service model consist of three entities the service provider, the service consumer, the service registry. The service provider creates or simply offers the web service. The service provider needs to describe the web service in a standard format, which in turn is XML and publish it in a central Service registry. The service registry contains additional information about the service provider, such as address and contact of the providing company, and technical details about the service. The Service Consumer retrieves the information from the registry and uses the service description obtained to bind to and invoke the web service. The appropriate methods are depicted in Figure 1 by the keywords ‘publish’, ‘bind’ and ‘find’.
Based on these protocols service components can be integrated into composite services at various levels to offer quality of services to satisfy user needs. In quality of service based composition abstract composition of services are given as input in form of languages, service locator finds appropriate services and after mediation & matchmaking of available web services list of alternative services generated for individual processes with different QoS attributes. The output requirement is to compose concrete service from set of alternative services based on service selection algorithms or criteria that satisfied user constraints.

In this paper we discuss some methods which are categorized into two categories: methods based on evolutionary or heuristic approaches such as Tabu search, Genetic algorithm, PSO, Basic local search etc. Non heuristic approaches or non-evolutionary includes linear programming, graph theory, knapsack problem, etc. In section 2 we discuss methods based on evolutionary approaches. Section 3 we discuss methods based on non-evolutionary approaches and finally conclusion of paper ends with comparing these categories in section 4.
Fig 3: Composition methods for heuristic and non-heuristic approaches

2. Methods based on Evolutionary approaches

Evolution approaches are generic population based meta heuristic optimizations approaches. They use mechanisms such as crossover, mutation, selection, and recombination. As discussed in figure 3 Heuristics approaches are divided into Meta heuristics and convex hull, further Meta heuristics is divided into Trajectory based methods which involve basic local search, simulated annealing, neighborhood search and Tabu search. Meta heuristic is an iterative generation process which guides subordinate heuristic by combining different concepts for exploring and exploiting the search space to provide optimal solutions. Basic local search is an iterative improvement to generate optimal solution until final conclusion not achieved. Kirkpatrick et al. [1] Proposed simulated annealing techniques in which random solution adopted with iterative improvement, if that solution results better than existing one, then accepts random solution as new solution with different probability. Glover [2] proposed idea of Tabu list from which best solution is opted and update Tabu list every time until condition met with desired solution for specific Tabu search. Hansen et al. [4] provides idea of variable neighbor- hood search to facilitate searching among neighbor variables and involves shaking, local search and movement of variables within search space. Further Meta heuristic also categorized into population based methods in which Genetic algorithm comes into role developed by Holland [3] on basis of principles of natural selection in which stronger individuals are winners in a competitive environment. GA operates on population of individuals, each individual represents possible solution to optimization problem. Individuals are evaluated depending upon fitness function which indicates how well an individual of population can solve optimization problem. Various genetic operators are used to achieve optimization solution. Fitness functions are used to maximize QoS attributes and to minimize some attributes by applying weight factor to every attributes. Ant colony optimization proposed by Dorigo [5] in which each ant applies a state transition rule to incrementally build a solution and applies a local updating rule to the pheromone or group until each of all ants has builds a complete solution. Global pheromone
updating rule is applied to yield optimal solution for optimization problem. Kennedy and Eberhart [6] proposed particle swarm optimization in which each particle keeps track of its highest fitness position in hyperspace. Individual populations are initialized by assigning random position and velocities due to which solution obtained to satisfy optimization problem needs. Velocities are modified on basis of individual population and best group population. In immune system [7] problem is encoded into an antibody. In immune selection, an antibody is suppressed in order to control their density in mating pool and make antibodies which not destroyed vaccines. Clonal operation used antibodies with high fitness as heuristic information for speeding up convergence.

3. Methods based on Non-Evolutionary approaches

In this section we discuss several non-evolutionary methods for service composition based on QoS attributes. Non heuristics methods are classified into several types such as Integer programming, Dynamic programming, Graph theory, Backtracking, Knapsack, Queens Problem etc. Integer programming is a mathematical optimization program in which all of variables are restricted to be integers whereas variants of integer programming known as linear programming deals with objective functions in which constraints are linear. Dynamic programming is a method in which optimization strategy is based on splitting the problem into smaller sub problems. Graph theory model service composition to obtain shortest path problem. In this method each service candidate represent node. They move QoS parameter from node to corresponding edges to construct direct acyclic graph. Chen, H. et al. [11] designs QoS capable QCWS architecture that implements QoS broker between web service clients and broker. Yu, T. et al. [17] propose different approaches for solving problem, first method for selection known as combinatorial approach and second approach on knapsack problem in which each item has weight, profit and knapsack has capacity. Algorithm is to select an item from each service class such that total utility under constraint is maximized. They consider multiple QoS constraint for service composition which uses different form of workflow in business process with help of BPEL languages etc. Zeng et al. [12] presents Agflow platform that enable quality driven composition with attributes such as execution price, duration, reputation and availability. SAW [13] technique used to achieve local optimization and global planning to compute overall QoS of each web service to select optimal one. Marganiec et al. [15] propose backward context based service selection method for service selection and composition in run time, it works as backtracking in which after selection of service at each step, algorithm goes back and check if selected service from previous steps are best for composition or not and then it invokes them. Alrifail et al. [16] propose algorithm that decompose main problem into sub problems and by solving these optimal solution is achieved. They preferred aggregation functions for computing QoS attributes and computes local constraints from global constraints which involves for service selection. Non heuristics approaches finds best and optimal solution but they lead to high complexity.
<table>
<thead>
<tr>
<th>Author name</th>
<th>Approach Used</th>
<th>Tools/Languages/models</th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Osman et al. [8]</td>
<td>Basic local search</td>
<td>Trajectory model</td>
<td>Iterative improvement on generated possible solution</td>
<td>Time consuming and increase overhead of maintaining possible solutions</td>
</tr>
<tr>
<td>Kirkpatrick et al. [1]</td>
<td>Simulated annealing</td>
<td>Probability based model</td>
<td>If random solution provides feasible result then assign it as an optimal solution with probability</td>
<td>Make system composition complex on involvement of random solutions</td>
</tr>
<tr>
<td>Glover et al. [2]</td>
<td>Tabu search</td>
<td>Tabu list</td>
<td>Intelligent exploration of neighbor-hood solutions to yield simplicity and effectiveness</td>
<td>Tabu tenure leads to costlier solution of problem if list length increased</td>
</tr>
<tr>
<td>Canfora et al. [9]</td>
<td>Genetic algorithm for composition</td>
<td>Fitness function and genetic operators</td>
<td>Fitness function maximize and minimize QoS attributes to yield user needs</td>
<td>Genetic operations are too complex to handle with user define weights and caused penalty</td>
</tr>
<tr>
<td>Jiuyen et al. [7]</td>
<td>Immune system algorithm</td>
<td>Immune and clonal tools</td>
<td>Antibodies with high fitness speed up convergence and accuracy to make vaccine</td>
<td>Composition increases infatuation among antibodies to destroy vaccine</td>
</tr>
<tr>
<td>Dorigo and Berbner [5]</td>
<td>Ant colony optimization</td>
<td>State transition rules based model</td>
<td>Composition problem solved by applying local updating rule to ant group</td>
<td>Updating rule applied repeatedly which increases cost and overhead</td>
</tr>
</tbody>
</table>
Chen, M. et al. [10]  | Particle swarm optimization  | Velocity & Position based model  | Each particle is a solution and has position and velocity to solve composition problem by identifying best position of element  | Every time velocity need to be modified to calculate best solution which decreases efficiency  

Table 2. Comparative view of Non Heuristic approaches

<table>
<thead>
<tr>
<th>Author name</th>
<th>Approach Used</th>
<th>Tools/Languages/models</th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chen et al. [11]</td>
<td>QCWS</td>
<td>QoS model</td>
<td>Composite services managed easily by clients with help of broker</td>
<td>Increases complexity to yield optimal solution</td>
</tr>
<tr>
<td>Yu, T et al. [17]</td>
<td>Combinatorial and Knapsack</td>
<td>QoS Broker based model</td>
<td>Response time and reliability of service increased</td>
<td>Cost increases under global constraint to maximize capacity</td>
</tr>
<tr>
<td>Zeng et al. [12]</td>
<td>Local optimization and global planning</td>
<td>AgFlow tool</td>
<td>Composition of web service done by reducing execution rate and price, availability</td>
<td>Workflow increased complexity based on user requirements</td>
</tr>
<tr>
<td>Marganiec et al. [15]</td>
<td>Backward context based service selection</td>
<td>Java</td>
<td>Fault tolerant and select next best service for available tasks</td>
<td>High overhead of backtracking service</td>
</tr>
<tr>
<td>Alrifai et al. [7]</td>
<td>Composition based approach</td>
<td>No specific</td>
<td>Decomposition of larger problem into sub problem facilitates optimal solution</td>
<td>Global constraint decomposition to local constraint is costlier</td>
</tr>
</tbody>
</table>
4. Conclusion

Service composition is a NP hard problem, but we need to find best solution for composite services. In this paper we studied several methods in service composition and categorize them into two categories, methods based on evolutionary approaches or heuristics and methods based on non-heuristics approaches. Evolutionary approach definitely finds a best optimal solution but complexity is more as new heuristics need to improve their speed up, whereas in non-evolutionary approaches time required to solve problem is high also final solution achieved is not necessary to be optimal. So different QoS attributes method discussed here that enables business process easier and distribute service candidates to achieve best solution in reasonable time.

References


