A Survey of QoS-aware Web Service Composition Techniques

Umar Shehu
Dept. of CST
University of Bedfordshire
Bedfordshire, UK

Gregory Epiphaniou
Dept. of CST
University of Bedfordshire
Bedfordshire, UK

Ghazanfar Ali Safdar
Dept. of CST
University of Bedfordshire
Bedfordshire, UK

ABSTRACT
Web service composition can be briefly described as the process of aggregating services with disparate functionalities into a new composite service in order to meet increasingly complex needs of users. Service composition process has been accurate on dealing with services having disparate functionalities, however, over the years the number of web services in particular that exhibit similar functionalities and varying Quality of Service (QoS) has significantly increased. As such, the problem becomes how to select appropriate web services such that the QoS of the resulting composite service is maximized or, in some cases, minimized. This constitutes an NP-hard problem as it is complicated and difficult to solve. In this paper, a discussion of concepts of web service composition and a holistic review of current service composition techniques proposed in literature is presented. Our review spans several publications in the field that can serve as a road map for future research.

Keywords
QoS, Service composition, Web service.

1. INTRODUCTION
Over the years, there has been a gradual shift in paradigm from an Internet based on physical computing networks to an Internet based on web services. This is in line with the long term vision of the Internet in which imagines web services are accessible anywhere, available everywhere, pervasive, and transparent [8]. The Internet is gradually getting closer to this vision with the discovery of web services. A web service can be defined as an Internet-accessible object that provides pre-defined capabilities [1] [2] [4]. Web service is generally loosely coupled and is a standard for integrating heterogeneous business applications with the aim of meeting growing consumer needs. For instance, a web service can allow people to perform hotel booking online. In many situations, it is possible that a given web service might not be able to satisfy more complex user requests. For example, suppose a user wants to plan a whole trip encompassing booking flight ticket, booking hotel room and processing payment, and he/she demands that the cost of executing web service(s) be minimized. It is impossible for a single web service to complete such a complex task as it includes several sub tasks (such as booking a hotel room, booking a flight ticket, and processing payment) and a QoS requirement or constraint such as minimum execution cost that must be satisfied before the overall request is successfully dealt with. In other words, the request will require a number of cheapest web services (payment processing service, hotel reservation service and airline reservation service) to be executed successively before the user is satisfied. In such situations, service composition is necessary. It combines several web services in order to build a composite service [3] [4] that is viewed by the user as a single web service and satisfies the request and QoS requirement. Service composition in the scenario described above becomes as straight forward as finding the composite service with the cheapest execution cost. However, if the user specifies an additional QoS constraint like response time should be minimized in addition to minimum cost, the service composition problem becomes complex as the best composite service would have to be the cheapest to execute and have lowest response time. Finding such a perfect solution in a large network of web services can be very difficult and time consuming because there will be plenty possible solutions to choose from in little amount of time, also solutions might have the best execution price but worst response time, or vice versa. As such, the problem is regarded as NP-hard [42]. Figure 1 further illustrates the previous example. After a request such as *Plan Journey* and QoS requirements like *minimum cost and response time* are made, the user is presented with a sequence of tasks that need to be executed to complete the request. In this case *Online Payment Offers*, *Credit Offers*, *Hotel Reservation,* and *Airline Reservation* are tasks to be performed via web service calls in order to serve the user request. For each of these tasks, there are lots of alternative web services (candidate services) capable performing them e.g. A1, A2 and A3 are the candidate services capable of completing the *Online Payment Offers* task. One web service is picked for each task according to its capability and QoS, and then orchestrated with web services selected from other tasks. Here, QoS is a metric that estimates how well a service satisfies the user. The output of the service composition process is a composite service whose QoS satisfies the user request and is a combination of the QoS of its individual candidate services. Because candidate services exhibit different QoS values, it has become a problem to compose only candidate services that will yield a composite service with the best QoS (in this case the cheapest execution price and minimum response time). In this paper, a comprehensive survey of techniques that tackle the NP-hard problem is presented. The paper starts with a discussion of the concepts of QoS-aware web service composition, focusing on QoS properties, workflow model and QoS aggregation functions. Then a review of existing approaches is presented and categorized into static and adaptive techniques. Finally, a conclusion terminates the paper.

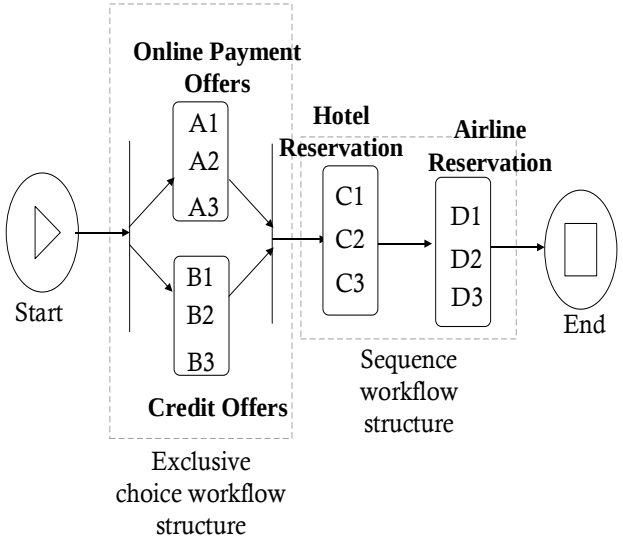

Figure 1: Travel booking scenario using service composition

2. BACKGROUND

2.1 QoS Properties

There are lots of web services that exist over the Internet, some having distinct functions (e.g. credit card service for online payment processing, airline booking service for buying flight tickets online etc.) while others have similar functions (e.g. master card service and visa card service are used for initiating online payments). Those having similar capabilities are distinguishable via their QoS values. QoS values determine whether a web service is reliable, trustworthy, or efficient. Its significance stems from the fact that a web service may be functionally capable of performing a given task, but might not reliable or efficient enough in performing the task up to the user's satisfaction. Web services are usually advertised with multiple QoS values, each value representing a quality aspect of the web service called a QoS property. QoS properties are generally the most commonly used characteristics in measuring the quality of web services or even composite services, this is because they indicate whether a service is capable of measuring up to user's expectations[9]. Some examples of common QoS properties include price, response time, reliability, reputation, encryption level etc.

Several classifications of web service QoS properties have been observed in literature. Authors in [33] classify QoS properties as *user dependent* and *user independent*. User dependent QoS properties are those properties whose values are different for each user. Examples of such properties include throughput, response time, etc. On the other hand, user independent properties are those properties that are similar for all users. QoS properties in this category include popularity, price, etc. A more tangible categorization is introduced in [30]. The authors made a distinction of QoS properties between *measurable* or *immeasurable*. Measurable QoS attributes are quantifiable using a QoS metric such as average execution time, while immeasurable QoS attributes are naturally qualitative e.g. flexibility, reputation, etc. QoS properties have also been classified as application and network parameters [34]. The former are associated with the web service application itself (e.g. availability, reliability, cost, etc.), while the latter are network related parameters that influence communication between a web service and the outside world. Authors in [34] identify three distinct network QoS parameters; Network delay, delay variation, packet loss. Network delay is the average amount of time it takes network packets to move across the network. Delay variation is the difference in the amount of time it takes each packet in a sequence of packets to reach its destination. Packet loss is the percentage of packets not delivered to destination after a group of packets has been sent across the network. Table 1 summarizes the different classifications of QoS properties.

Table 1. Classification of major QoS properties

QoS Property	Measurable	Immeasurable	User dependent	User independent	Network level	Application level
Throughput	✓		✓			✓
Response time	✓		✓			✓
Cost	✓			✓		✓
Reliability	✓			✓		✓
Network delay	✓			✓	✓	
Reputation		✓	✓			✓

2.2 QoS -aware web service composition

Service composition aggregates a set of web services whose capabilities can be discovered spontaneously in order to meet an acceptable outcome. The composition process is akin to the integration process of workflow management systems [6]. A Workflow management system functions according to a pre-defined workflow model, which consist of tasks and transitions [13]. In the area of service composition, a workflow model is used for the aggregation of QoS properties. This is achieved by creating abstract descriptions which select and compose existing services to form workflows [11]. Workflow is a sequence of tasks to be performed in order to achieve a set goal. It is typically modeled in different structures. The main forms of workflow structures are sequence, choice, parallel split and loop.

According to [10] and [12], the web service composition process involves a series of steps as shown in figure 2. Firstly, the user's request (abstract task) is determined and subsequently decomposed into a sequence of sub-tasks. The sequence of sub-tasks is then further broken down into workflow tasks using a standard web service framework like WS-BPEL. For each workflow task, web services are discovered and classified into service classes according to which tasks they can implement, every service class representing a group of web services with similar capabilities. They are mapped as one service class per workflow task. Once the classification is done, a web service is selected from each service class and bound to a composite service. In environments where number of web services per workflow task is large, a lot of composite services that will be formed as result. Therefore, the last step involves the selection of the best composite service that maximizes (or minimizes) a given QoS property as specified by the user request.

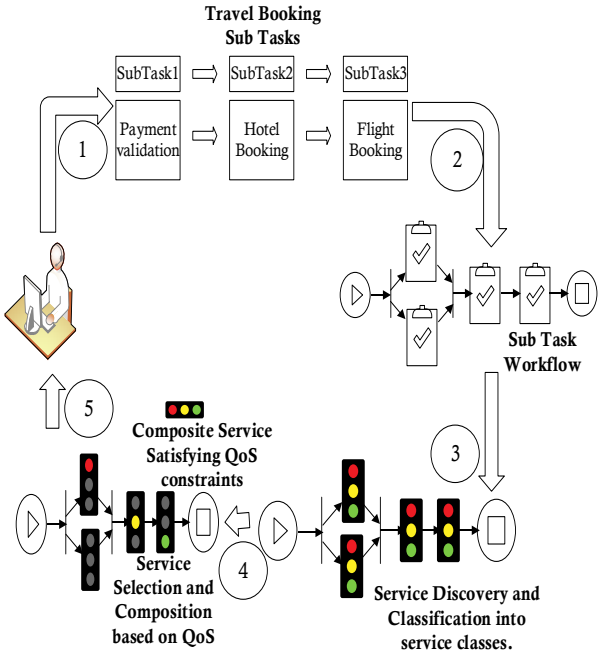

Figure 2 QoS-aware service composition process.

2.3 QoS Aggregation

The QoS value of a composite service is calculated by aggregating QoS of candidate services based on the type of workflow structures involved. For instance, considering the workflow for the travel booking example shown in Figure 1, assuming the user requires minimization of cost and response time QoS. Also assuming that web services A1,C2 and D1 have been selected to form a composite service (A1-C2-D1). Then the end-to-end total response time T_{RT} of this composite service will be the sum of response times of A1 (including probability of choosing A1), C2 and D1.

$$RT_T = \left[p_{A1} \times RT_{A1} \right] + RT_{C2} + RT_{D1} (1)$$

Where p_{A1} is the probability of choosing A1 in the exclusive choice workflow structure. RT_{A1}, RT_{C2}, and RT_{D1} are response time QoS values for A1, C2 and D1 web services respectively. A similar formula can be used in the case of finding total cost of the composite service.

$$C_T = \left[p_{A1} \times C_{A1} \right] + C_{C2} + C_{D1} (2)$$

Where C_{A1}, C_{C2}, and C_{D1} are execution cost QoS values for A1, C2 and D1 web services respectively. Table 2 shows the QoS aggregation formulas for sequence, parallel, loop and exclusive choice workflow structures.

3. CURRENT APPROACHES

QoS-aware service composition problem is describe as an NP-hard problem[49]. Several techniques have been developed to tackle QoS-aware service composition problem. Some have deal with service composition problem under non-varying QoS environment, while others have dealt with the problem in varying QoS environments (runtime environments). We therefore classify current approaches according to whether not they take QoS changes into account. This classification is necessary as it is currently missing in literature. In this section, techniques are classified into static and adaptive approaches.

Table 2. Aggregation formulas for QoS computation of composite service.

QoS property	Sequence structure	Parallel structure	Loop structure	Exclusive choice structure
Response time	$\sum_{i=1}^{n} RT(S_i)$	$Min\left(RT(S_i)\right)$	$k.RT(S)$	$\sum_{i=1}^{n} p_i * RT(S_i)$
Reputation	$\prod_{i=1}^{n} RP(S_i)$	$Max\left(RP(S_i)\right)$	$RP(S)^k$	$\prod_{i=1}^{n} p_i * RP(S_i)$
Cost	$\sum_{i=1}^{n} C(S_i)$	$Min\left(C(S_i)\right)$	$k.C(S)$	$\sum_{i=1}^{n} p_i * C(S_i)$
Reliability	$\prod_{i=1}^{n} R(S_i)$	$Min\left(R(S_i)\right)$	$R(S)^k$	$\prod_{i=1}^{n} p_i * R(S_i)$

3.1 Static approaches

This category of techniques perform web service composition using prior knowledge of web service QoS values. Also, they do not consider dynamic changes in QoS. Hence, they are not ideally suitable in runtime environment where web service QoS values are constantly changing over time. We further classify techniques here into 4 categories namely; (1) Local maximization approaches; (2) Linear optimization approaches; (3) Approximation approaches and (4) Pareto-optimization approaches.

3.1.1 Local maximization approaches

One of the reasons why there is difficulty in dealing with a QoS service composition problem is the presence of both local QoS constraints (applied to each web service) and global QoS constraints (applied to the whole composite service). Ideally the solution would have to satisfy both of these requirements before being called an optimal solution, which can be a difficult to achieve. A method current studies have used in reducing complexity of the problem is to consider only local constraints in selecting best candidate services for each sub task. This way, only local constraints will need to be satisfied before solution is executed. The process is known as local maximization. For example, instead of finding the composite service that has total response time less than 20ms, local maximization finds the composite service whose candidate services each have response times less than 20ms. In this case, the best solution may not necessarily have total response time less than 20ms. Using the method, an optimal solution can be reached in little amount of time. A popular local approach is Dynamic Programming[17][18] which breaks down an execution plan into separate divisible and indivisible parts. It computes an optimal solution for each divisible part while relying on a recursive branch-and-bound algorithm in finding optimal solutions for the indivisible parts. Another local approach is a technique proposed in [15]. The technique uses a learning-depth-first search (LDFS) algorithm that combines bound depth first searches with learning iteratively. Some other technique for local maximization is one based on Simple Additive Weighing(SAW)[48]. This method assigns scores to each candidate services by

multiplying their QoS values with a user-defined weight value. The weight value signifies the importance of satisfying a given QoS constraint specified by the user. The method then selects the web service with the best score for each task. An advantage of local approaches is that their running time scales well with an increase in number of services. However, they suffer from loss of accuracy as a result of the pre-selection process involved.

3.1.2 Linear optimization approaches

Linear approaches refine the NP-hard problem by assuming a linear objective function. Objective function is a measure of how good the QoS value of a composite service is. It is determined by combining QoS property values of all the candidate services contained in the composite service into a single aggregate value. Linear optimization of web service composition is achieved using Linear Integer Programming (LIP) techniques [19][18]. LIP can find the best composite service without necessarily building all possible compositions. It can also integrate both functional and non-functional service properties [16] into the composition process. LIP is usually useful when dealing with a small set of services. Although, it can take a lot of time to find the best composition in large service environments.

3.1.3 Approximation approaches:

Current studies have also focused on techniques for finding approximate solutions as they are faster to reach than optimal ones. These approaches are usually heuristic in nature as they achieve optimization by performing various computations and iterations on possible solutions with respect to a given degree of quality. Heuristics are capable of arriving at solutions while making little or no assumptions about the problem. They are also capable of rigorously covering very large search spaces in relatively small amount of time. Several approximation approaches have been introduced. One popular approach is Particle Swarm Optimization (PSO) technique. PSO simulates the natural behavior of birds when searching for food particles, each particle representing a possible solution (composite service). All particles within the swarm are assigned parameters such as position(X), velocity (V), best local position ($\underline{Xbi}$), and best global position ($\underline{Xgi}$). Ming et al. [19] first investigated the application of Particle Swarm Optimization (PSO) in the context of service composition. They introduced a discrete PSO approach called DPSO to finding approximate solutions. Their technique uses best global position ($\underline{Xgi}$) to update location and speed parameters for all particles in the swarm after each iteration;

$$X_{i+1} = X_i + V_{i+1} \quad (3)$$

$$V_{i+1} = V_i + C1R1(Xb_i - X_i) + C2R2(Xg_i - X_i) \quad (4)$$

Where vectors $\underline{X_i}$ is a particle position, $\underline{V_i}$ represents particle velocity, $\underline{Xb_i}$ represents particle's best recorded position, $\underline{Xg_i}$ is the global best particle position, C1 and C2 are called social parameters, while R1 and R2 are randomly generated numbers between 0 and 1. The technique finds good quality solutions in considerable amount of time, but suffers from been trapped into local optima more often than not. An improved PSO approach is presented in [20]. This approach incorporates adaptive mutation and learning factor ability into the basic PSO technique. Here, adaptive mutation mechanism prevents particles from being trapped in the local maxima by letting them hop out during early stages of computation. While learning factor tunes C1 and C2 values so as to improve population density of particles during computation.

Another popular approximation technique is one based on Genetic Algorithms (GA). GAs are capable of evolving members of a generation (genomes) according to a set of rules up to a point where fitness value is optimized. Each genome can represent a possible composite service and is usually encoded in form of array of numbers. GA initiates the optimization process by building an initial generation of genomes which are subsequently to find the ones with best fitness values. These best individuals are then placed into a mating pool where they are altered by crossover and mutation operators in order to improve their fitness. Canfora et al. [42] studied how GA can be used in solving the service composition problem. In their approach they encode candidate services (WS_{ij}) in form of an integer array inside each gene of a genome (as seen in Figure 3). The aim of their technique is to find a combination of array elements that achieves the best fitness value while meeting QoS constraints. Their approach find good quality solutions most of the time, however it often traps into local optima. Authors in [54] present an improved GA that reduces the likelihood of trapping into local optima. They present two functions; an improved mutation function controls mutation process via mutation probability parameter, The other function is a partial initialization function that stores set of best genomes which will be later used in performing crossover with other genomes.

Literature indicators suggest that approximation approaches are more efficient than linear approaches as they can rapidly eliminate large numbers of possible execution plans in a relatively small amount of time. Furthermore, they can handle large number of services better than linear methods. However, they suffer from lack of ability to find optimal solutions most of the time. It is possible that a best solution was discarded during the elimination process using these approaches.

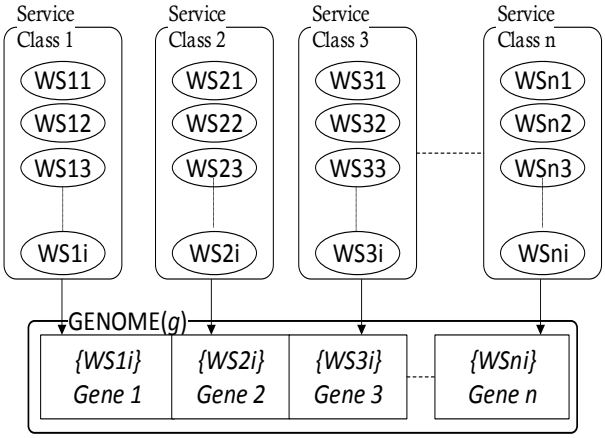

Figure 3 Service combination represented as a genome.

3.1.4 Pareto-optimization approaches

These approaches find solutions that are optimal with respect to one or more objective functions, but not all. One method that yields pareto solutions is a two-stage conversion of a multi objective problem into a single objective problem as proposed in [24]. This involves normalizing an objective function's range, and then using Simple Additive Weighting (SAW) technique to weight each objective function. In [25] a multi objective optimization algorithm that relies on cost of service failure to rank pareto-optimal solutions with respect to a decision maker's preferences is proposed. The algorithm, however, cannot distinguish which pareto-optimal solution is the better one. Study in [22] suggests a genetic algorithm based on non-dominated sort (NSGA-II) that map genomes to

their respective fronts in order to sort and compare pareto-optimal solutions. [23] Employs a different comparison technique based on binary quality indicators that analyze pareto-optimal solution sets. Pareto-optimization approaches have been observed to provide better performance and good quality solutions than other approaches.

3.2 Adaptive approaches

In the previous section, techniques covered that focus on finding optimal solutions to the NP-Hard problem in static environments i.e. environments where QoS values of web services are already known prior to generating the task workflow. Some current studies have extended the service composition problem to finding optimal solutions in situations where web service QoS values are not known prior to generating the workflow. Such examples reflect a more realistic scenario where runtime web service QoS values vary from values advertised by service providers. Adaptive approaches are capable of adapting to changes in QoS values of the service environment. Adaptive service composition is a very active area of research attracting much of attention in recent years. It is divided into two types [33]; *Internal composition adaptation* and *External composition adaptation* techniques. The former approach reacts to environmental changes by rebuilding a composition either from ground up or from the point of fault within the composite service. On the other hand, the latter approach uses adjustable adapters that bridge the gap between the service workflow and the dynamically changing service environment.

There are several internally adaptable service composition approaches, most of which have focused on small service environments. Amongst these approaches are AI Planning-based techniques [55][33]. In [55] the authors present an AI planning approach that relies on a moderately accurate Case Based Reasoning (CBR) technique to build service compositions on-the-fly. CBR is used to obtain solutions from a set of composition cases gathered from past experiences. If such solutions do not exist, then an AI planner builds composition solutions from ground up. The other study [33] presents a self-adaptive service composition that is based on AI planning graphs called Graph plan repair. Their approach reconfigures compositions during runtime with the aid of a greedy search algorithm used to explore the planning graph for possible service combinations that can achieve the user's goal. The algorithm scouts through the planning graph to find and repair services that don't meet user goal due to their unexpected change. If such services do exist, then new services are added into the graph to satisfy user goals. Afterwards, the whole process starts all over until all services satisfy user goals. The method presented in [38] uses AI planning to dynamically map user requirements to service workflows. The approach is goal-oriented, thus any changes to user requirements at run time will ultimately be applied to the workflow structure. The major drawback of AI planning-based techniques is that they do not perform well in situations where the number of participating web services is large. Also they are susceptible to sub-optimal solutions in such environments.

Several internal adaptation solutions focus on using Reinforced learning (RL) techniques to solve adaptive service composition problem. Work in [34] proposes an adaptive RL method based on Markov Decision Process (MDP) that finds optimal solution at runtime. MDP builds a model for obtaining compositions consisting of multiple aggregated workflows. RL method takes over in finding optimal solution (or pareto-optimal solutions) by acquiring MDP policy with the optimal QoS. Any change in service environment will prompt a change of MDP policy for the sake of continuing the learning process. In [26] an extended MDP method called Semi Markov Process (SMP) has been introduced to forecast QoS and network efficiency of web service environment during composition. The output from SMP then determines which web services need to be replaced as a result of poor QoS. In [41] the authors propose a method that re-plans composition once it predicts a difference between estimated QoS and runtime QoS values. In their approach, the authors utilize a proxy-based model to achieve runtime binding of web services. [35] Propose an improved RL approach that utilizes Reuse Strategy to enhance performance and stability of RL. Work has been extended in [36] which introduce a randomized RL technique that considers multiple QoS and non-QoS criteria like cost, reputation, deadline and user preferences to obtain optimal solutions while adjusting to runtime changes in availability of service environment. RL-based approaches enhance the service composition process with exploration and exploitation capabilities, making the composite services more reactive to environmental changes. However they impose high computational cost especially in large service environments. Additional studies [40] have modeled the adaptive service composition problem as a shortest path problem. Here, a shortest path algorithm (CSP) is used to ascertain a faulty web service and come up with an alternative path to a backup web service. This approach often traps into local optima and doesn't always produce the best results. [37] Focused on using heuristic approach to tackling adaptive service composition. Their approach makes use of a K-means algorithm to find pareto-optimal solutions. The algorithm works by firstly normalizing QoS constraints then a local classification is made to group candidate services into clusters with respect to their QoS levels, upon which a heuristic algorithm performs global optimal selection. Adaptation is performed by using a utility threshold τ to tune selection of clusters based on environmental constraints such as time or service density. With the aid of its classification technique, the approach performs efficiently in environment where number of web services is large.

Some *external composition adaptation* techniques have been proposed in recent studies. These techniques use composition methods, policies or protocols to continually regenerate and update optimal service workflows according to changes in environment. *Social network analysis* techniques have been recently introduced to tackle adaptation in service composition. They are methods used to map and measure the relationship between web services in a social network [51]. An example is proposed in [27] which applied link analysis and page ranking to rank services based on their success and failure popularity. In order to obtain such information, snapshots of the whole service registry are taken at regular intervals so that popularity changes can be reflected upon service workflows accessible to the user at runtime. A modified page rank approach namely *service rank* is presented in [27] which applied link analysis and page ranking to rank services based on their success and failure popularity. In order to obtain such information, snapshots of the whole service registry are taken at regular intervals so that popularity changes can be reflected upon service workflows accessible to the user at runtime. A modified page rank approach namely *service rank* is presented in [39]. In this approach, web services were ranked based on connectivity and invocation history. The method combines ranking score with QoS score for composition ranking. Social network analysis techniques are more efficient than internal adaptation approaches since

the information required for the adaptation process is obtained prior to the commencement of service composition. Its only drawback is that it requires more computational resources in order to acquire information that will be useful for adapting composite services. Table 3 summarizes the service composition techniques covered in this paper.

Table 3 - Classification of approaches for QoS-aware web service composition.

Categories	Sub-categories	Techniques
Static approaches	Local optimization approaches	Dynamic programming
		Learning depth first search
		Simple additive weighing
	Linear optimization approaches	Linear programming
	Approximation approaches	Genetic algorithm
		Particle swarm algorithm
	Pareto optimization approaches	Genetic algorithm
		Particle swarm algorithm
		Multi-objective simple additive weighing
Adaptive approaches	Internal adaptation approaches	AI-planning
		Reinforced learning
		Graph-based approach
		Heuristic algorithm
	External adaptation approaches	Protocol-based approach
		Social network analysis approach

4. CONCLUSION

In this paper, a comprehensive survey of QoS aware service composition techniques is provided. These techniques are classified into static and adaptive approaches. Static approaches are methods whose mechanisms are based on static predefined QoS values. They are further divided into local approaches, linear optimization approaches, approximation approaches and pareto-optimization approaches. Amongst these approaches, It is discovered that the later seems to be the most efficient of them especially in large service environments. Adaptive approaches on the other hand were further categorized into internal adaptation approaches and external adaptation approaches. External adaptation techniques being the more efficient, albeit at increased computational cost. Further research is required on these techniques to determine the extent of their usefulness in web service composition.

5. REFERENCES

[1] Alonso, G.; Casati, F.; Kuno, H.; Machiraju, V.;"Web services: Concepts, Architecture and Applications," *Springer Verlag (ISBN:3540440089) on*, vol., no., pp.124-125, June 2003.

[2] Singh, P.; Huhns, N.;"Service-Oriented Computing: Semantics, Processes, Agents," *John Wiley and Sons (ISBN: 0470091487) on*, vol., no., pp., January 2005.

[3] Casati, F.; Georgakopoulos, D.;,Proceedings of the international workshop on Technologies for E-Services Roma, Italy on, vol., no., pp., September 2001.

[4] Tsur, S.; Abiteboul, S.; Agrawal, S.; Dayal, U., Klein, J; Weikum, G.;,"Are web Services the Next Revolution in e-commerce?," *Proceedings of the International Conference on very large databases on*, vol., no., pp. 614-617, September 2001.

[5] Liangzhao Zeng; Benatallah, B.; Ngu, A.H.H.; Dumas, M.; Kalagnanam, J.; Chang, H.; , "QoS-aware middleware for Web services composition," *Software Engineering, IEEE Transactions on* , vol.30, no.5, pp. 311- 327, May 2004.

[6] Jinghai Rao; Xiaomeng Su;,"A Survey of Automated Web Service Composition Methods," In Proceedings of the First International Workshop on Semantic Web Services and Web Process Composition, SWSWPC on, vol., no., pp.1-12, 2004.

[7] Fang Liu; Bo Peng; , "Immune-Particle Swarm Optimization Beats Genetic Algorithms," *Intelligent Systems (GCIS), 2010 Second WRI Global Congress on* , vol.3, no., pp.233-236, 16-17 Dec. 2010.

[8] Kleinrock Leonard;, "An Internet Vision: the invisible global infrastructure," *Ad Hoc Networks on*, vol.1, no., pp.3-11, 2003.

[9] Shi Yulu; Chen Xi; , "A Survey on QoS-aware Web Service Composition," *Multimedia Information Networking and Security (MINES), 2011 Third International Conference on* , vol., no., pp.283, 4-6 Nov. 2011

[10] Strunk, A.; , "QoS-Aware Service Composition: A Survey," *Web Services (ECOWS), 2010 IEEE 8th European Conference on* , vol., no., pp.67, 1-3 Dec. 2010.

[11] Jaeger, M.C.; Rojec-Goldmann, G.; Muehl, G.; , "QoS aggregation for Web service composition using workflow patterns," *Enterprise Distributed Object Computing Conference, 2004. EDOC 2004. Proceedings. Eighth IEEE International* , vol., no., pp. 149- 159, 20-24 Sept. 2004.

[12] Kim, A.; Kang, M.; Meadows, C.; Loup, E.; Sample, J.;,"A Framework for Automatic Web Service Composition," *Naval Research Laboratory Center for High Assurance Computer Systems on* , vol.,no.,pp.595-598, 2009.

[13] Shi Yulu; Chen Xi; , "A Survey on QoS-aware Web Service Composition," *Multimedia Information Networking and Security (MINES), 2011 Third International Conference on* , vol., no., pp.284, 4-6 Nov. 2011

[14] Cardoso, J.; Sheth, A.P.; Miller, J.A.; Arnold, J., Kochut, K.;,"Quality of service for workflows and web service processes," *J. Web Sem. on*, vol.,no., pp.281-308, 2004.

[15] Wonhong Nam; Hyunyoung Kil; Jungjae Lee; , "QoS-Driven Web Service Composition Using Learning-Based Depth First Search," *Commerce and Enterprise Computing, 2009. CEC '09. IEEE Conference on* , vol., no., pp.507-510, 20-23 July 2009.

[16] Changlin Wan; Ullrich, C.; Limin Chen; Rui Huang; Jiewen Luo; Zhongzhi Shi; , "On Solving QoS-Aware Service Selection Problem with Service Composition," *Grid and Cooperative Computing, 2008. GCC '08. Seventh International Conference on* , vol., no., pp.467-474, 24-26 Oct. 2008.

[17] Bonet Blai;," Learning Depth-First Search: A Unified Approach to Heuristic Search in Deterministic and Non-Deterministic Settings, and its application to MDPs," *In Proceedings of ICAPS™, pp.3-23 2006.*

[18] Zhenqiu Huang; Wei Jiang; Songlin Hu; Zhiyong Liu; , "Effective Pruning Algorithm for QoS-Aware Service Composition," *Commerce and Enterprise Computing, 2009. CEC '09. IEEE Conference on* , vol., no., pp.519-522, 20-23 July 2009.

[19] Chen Ming; Wang Zhen wu;,"An Approach for Web Service Compositon Based on QoS and Discrete Particle Swarm Optimization," *Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, SNPD 2007, Eight ACIS International Conference on,* vol.2, no., pp. 37-41, August 2007.

[20] Lou Yuan-sheng; Hu Pa; Tao Fu-ling;,"An Improved Particle Swarm Optimization and its Application on Web Service Composition," *Computer Application and System Modelling (ICCASM), 2010 International Conference,* vol.11, no.,pp.V11-44-V11-47, October 2010.

[21] Yan Gao; Jun Na; Bin Zhang; Lei Yang; Qiang Gong; , "Optimal Web Services Selection Using Dynamic Programming," *Computers and Communications, 2006. ISCC '06. Proceedings. 11th IEEE Symposium on* , vol., no., pp. 365- 370, 26-29 June 2006.

[22] Seog-Chan Oh; Jung-Woon Yoo; Hyunyoung Kil; Dongwon Lee; Kumara, S.R.T.; , "Semantic Web-Service Discovery and Composition Using Flexible Parameter Matching," *E-Commerce Technology and the 4th IEEE International Conference on Enterprise Computing, E-Commerce, and E-Services, 2007. CEC/EEE 2007. The 9th IEEE International Conference on* , vol., no., pp.533-542, 23-26 July 2007.

[23] Shi Yulu; Chen Xi; , "A Survey on QoS-aware Web Service Composition," *Multimedia Information Networking and Security (MINES), 2011 Third International Conference on* , vol., no., pp.285, 4-6 Nov. 2011

[24] Zhifeng Gu; Bin Xu; Juanzi Li; , "Inheritance-Aware Document-Driven Service Composition," *E-Commerce Technology and the 4th IEEE International Conference on Enterprise Computing, E-Commerce, and E-Services, 2007. CEC/EEE 2007. The 9th IEEE International Conference on* , vol., no., pp.513-516, 23-26 July 2007.

[25] Yoo, J.J.-W.; Kumara, S.; Dongwon Lee; Seog-Chan Oh; , "A Web Service Composition Framework Using Integer Programming with Non-functional Objectives and Constraints," *E-Commerce Technology and the Fifth IEEE Conference on Enterprise Computing, E-Commerce and E-Services, 2008 10th IEEE Conference on* , vol., no., pp.347-350, 21-24 July 2008.

[26] Rami Mounla;,"QoS-aware Web Service Composition," Computer Science University of Auckland on, vol., no.,pp.116-122,2008.

[27] Strunk, A.; , "QoS-Aware Service Composition: A Survey," *Web Services (ECOWS), 2010 IEEE 8th European Conference on* , vol., no., pp.70, 1-3 Dec. 2010.

[28] Yu Dai; Lei Yang; Bin Zhang; "QoS-driven self-healing web service composition based on performance prediction," *J. Comput. Sci. Technol.on,* vol.24,no.2, pp.250-261, March 2009.

[29] Hongbing Wang; Xiaohui Guo; , "An Adaptive Solution for Web Service Composition," *Services (SERVICES-1), 2010 6th World Congress on* , vol., no., pp.503-510, 5-10 July 2010.

[30] Wang, L.; Shen, J.; Yong, J;," A survey on bio-inspired algorithms for web service composition," *In Proceedings of CSCWD on*, vol., no., pp.569-574, 2012.

[31] Fouad, H.; Baghdad, A.;,"Dynamic Web Service Composition: Use of Case Based Reasoning and AI Planning," In Proceedings of ICWIT on, vol., no., pp.22-29, 2012

[32] Huipeng Guo; Jianxin Li; Zongxia Du; Mu Li; , "PAAS: A Protocol-based Approach to Adaptive Service Composition," *Computer Application and System Modeling (ICCASM), 2010 International Conference on* , vol.4, no., pp.V4-601-V4-605, 22-24 Oct. 2010.

[33] Zibin Zheng; Yilei Zhang; Lyu, M.R.; , "Distributed QoS Evaluation for Real-World Web Services," *Web Services (ICWS), 2010 IEEE International Conference on* , vol., no., pp.83-90, 5-10 July 2010.

[34] KangChan Lee. 2003. QoS for Web Services: Requirements and Possible Approaches. [ONLINE] Available at: http://www.w3c.or.kr/kr-office/TR/2003/ws-qos. [Accessed 12 February 13].

[35] Yuhong Yan; Poizat, P.; Ludeng Zhao; , "Self-Adaptive Service Composition Through Graphplan Repair," *Web Services (ICWS), 2010 IEEE International Conference on* , vol., no., pp.624-627, 5-10 July 2010.

[36] Hongbing Wang; Xuan Zhou; Xiang Zhou; Weihong Liu; Wenya Li; , "Adaptive and Dynamic Service Composition Using Q-Learning," *Tools with Artificial Intelligence (ICTAI), 2010 22nd IEEE International Conference on* , vol.1, no., pp.145-152, 27-29 Oct. 2010.

[37] Qing Liu; Yulin Sun; Shilong Zhang; , "A Scalable Web Service Composition Based on a Strategy Reused Reinforcement Learning Approach," *Web Information Systems and Applications Conference (WISA), 2011 Eighth* , vol., no., pp.58-62, 21-23 Oct. 2011.

[38] Jureta, I.J.; Faulkner, S.; Achbany, Y.; Saerens, M.; , "Dynamic Web Service Composition within a Service-Oriented Architecture," *Web Services, 2007. ICWS 2007. IEEE International Conference on* , vol., no., pp.304-311, 9-13 July 2007.

[39] Nebil Ben Mabrouk; Sandrine Beauche; Elena Kuznetsova; Nikolaos Georgantas; Valerie Issarny;, "QoS-aware service composition in dynamic service oriented environments," In *Proceedings of the 10th ACM/IFIP/USENIX International Conference on Middleware* (Middleware '09). Springer-Verlag New York, Inc., New York, NY, USA on, vol., no.7, pp.1-20.

[40] Zhanlei Ma; Lin Liu; Hongji Yang; Mylopoulos, J.; , "Adaptive Service Composition Based on Runtime Requirements Monitoring," *Web Services (ICWS), 2011 IEEE International Conference on* , vol., no., pp.339-346, 4-9 July 2011.

[41] Adrian Klein; Fuyuki Ishikawa; Shinichi Honiden;,"Towards network-aware service composition in the cloud," In *Proceedings of the 21st international conference on World Wide Web* (WWW '12) on, vol., no., pp.959-968, 2012.

[42] HaiTao Song; Yanming Sun; Yingyu Yin; Shixiong Zheng; , "Dynamic Weaving of Security Aspects in Service Composition," *Service-Oriented System Engineering, 2006. SOSE '06. Second IEEE International Workshop* , vol., no., pp.189-196, Oct. 2006.

[43] Ponnalagu, K.; Narendra, N.C.; Krishnamurthy, J.; Ramkumar, R.; , "Aspect-oriented Approach for Non-functional Adaptation of Composite Web Services," *Services, 2007 IEEE Congress on* , vol., no., pp.284-291, 9-13 July 2007.

[44] Ludwig, S.A.; , "Clonal selection based genetic algorithm for workflow service selection," *Evolutionary Computation (CEC), 2012 IEEE Congress on* , vol., no., pp.1-7, 10-15 June 2012.

[45] Liangzhao Zeng, Boualem Benatallah, Marlon Dumas, Jayant Kalagnanam, and Quan Z. Sheng. Quality driven web services composition. In *Proceedings of the 12th international conference on World Wide Web* (WWW '03). ACM, New York, NY, USA, pp.411-421, 2003.

[46] J. Liang and K. Nahrstedt;, "Service Composition for Advanced Multimedia Applications," vol. 1, no. c, pp.1-13.

[47] Jaeger, M.C.; Muhl, G.; Golze, S.;, "QoS-aware composition of Web services: a look at selection algorithms," *Web Services, 2005. ICWS 2005. Proceedings. 2005 IEEE International Conference on* , vol., no., pp.,808, 11-15 July 2005.

[48] Bansal,S.; Bansal, A.; Blake, M.;,"Trust-based Dynamic Web Service Composition using Social Network Analysis," *Arizona State University* vol., no., pp.1-8, Dec 2010.

[49] Amiri, M.A.; Serajzadeh, H., "QoS aware web service composition based on genetic algorithm," *Telecommunications (IST), 2010 5th International Symposium on* , vol., no., pp.502,507, 4-6 Dec. 2010.

[50] Henni, F.; Atmani, B.;,"Dynamic Web Service Composition. Use of Case Based Reasoning and AI Planning," *in Mimoun Malki; Salima Benbernou; Sidi Mohamed Benslimane & Ahmed Lehireche, ed., 'ICWIT' , CEUR-WS.org,* , vol., no., pp. 22-29 . 2012.