International Journal of Advanced Trends in Engineering and Technology (IJATET)
Impact Factor: 5.665, ISSN (Online): 2456 - 4664
(www.dvpublication.com) Volume 2, Issue 2, 2017

MODIFIED PSO BASED QUALITY ENHANCED SERVICE SELECTION AND ORCHESTRATION

Dr. K. Adlin Suji* & Dr. S. Sujatha**
* Faculty, Department of Computer Applications, National Institute of Technology, Trichy, Tamilnadu
** Associate Professor, Department of MCA, BIT Campus, Anna University, Chennai, Tamilnadu

Cite This Article: Dr. K. Adlin Suji & Dr. S. Sujatha, "Modified PSO Based Quality Enhanced Service Selection and Orchestration", International Journal of Advanced Trends in Engineering and Technology, Volume 2, Issue 2, Page Number 118-125, 2017.

Abstract:

The interconnected world has simplified several aspects of a processing system. The best of them being component reuse. Software components written by professionals can be reused in a system to incorporate required functionalities. Web services are such components facilitating reuse. However, due to the huge number of web services available in the repository, the process of selecting an appropriate service for the current requirement becomes challenging. The selection requirements do not demand best services, instead, they require optimal services. This paper presents a metaheuristic based solution for service selection and service optimization. Modified PSO is used for the selection and orchestration process. PSO is discretized and incorporated with catfish particles to eliminate the problem of local optima. It was observed from the experiments that the proposed modified PSO based service selection and orchestration performs effectively in terms of optimal selection and time.

Key Words: Service Selection, Service Orchestration, PSO & Optimization

1. Introduction:

Web services are computational components designed to build service oriented distributed systems [1]. Increase in such architectures have led to an increased number of web services providing similar functionalities. However, distinction within them is brought about by the quality levels provided by the services. Customer's satisfaction levels while using these services are determined solely by the Quality of Service (QoS) of the service [2, 3]. However, it is not necessary to provide the best available service to every user. Performing this type of service assignments will not only underutilize the capabilities, they will also increase the cost for the user. Service requests are usually paired with the QoS requirements, and it is sufficient to allocate a service that satisfies the user's requirements [4, 5]. Service selection and orchestration are usually performed keeping a single user in mind. However, the overload of web services is not considered. Other issues in this section includes missing QoS requirements. The selection schemes are usually proposed considering that the QoS requirements are complete. In real time, it is not so. This paper proposes a metaheuristic based flexible technique that can operate on QoS requirements to provide effective solutions.

2. Related Works:

Service selection and orchestration have become major necessities for the current systems. This section presents recent techniques that have been proposed to perform service selection and orchestration. A technique that considers multiple users while performing service selections is presented by Wang et al. in [6]. This technique is also proposed to handle missing QoS values to provide optimal solutions. Other techniques considering multiple users during the selection phase are [7, 8], proposed by Shahand et al. and Dayachunk et al. Fuzzy based service selection technique that performs service query optimization is presented by Chouiref et al. in [9]. This technique uses a priority based scheme to aggregate elementary similarities and the top-k results are passed to the user for final selection. A case based reasoning technique that identifies similarity between components to identify the appropriate web services is presented by Renzis et al. in [10]. This technique uses three different similarity function for analysis and uses case based reasoning for processing. Other similar techniques for appropriate service selection includes [11, 12]. A Social Spider algorithm based service selection technique is proposed by Mousa et al. in [13]. This technique was proposed with the main concern for reducing the time consumption during the selection process. Several heuristic based techniques were proposed to tackle the service selection issue. Some of them include [14, 15, 16], however they tend to tradeoff time or optimization for the benefit of the other. A multi agent based distributed optimization technique for QoS based service selection is proposed by Temglit et al. in [17]. This technique uses an algorithm called Synch4QoS that operates on a multi-user architecture to provide real-time service selection. A Mixed Linear Programming (MLP) based optimal component selection technique was proposed by Zeng et al. in [18]. However, this technique has scalability issues. Genetic Algorithm (GA) has been a major favourite heuristic for this process. Studies incorporating Ga includes [19-23]. A context aware QoS based service selection system was proposed by Xu et al. in [24]. User preference based techniques for service selection are currently on the raise, includes [25, 26].

3. Modified PSO based Quality Enhanced Service Selection and Orchestration:

The process of service selection and orchestration has acquired the NP-Hard status due to the hugeness of the data involved and the complexity of the operations. Hence optimizing this process using metaheuristics can provide a very effective solution in terms of both efficiency and time. This work adapts Catfish PSO (C-PSO), discretizes it and proposes an algorithm to effectively perform service selection and service orchestration in a large service repository. The process of service selection and orchestration are described as two phases. Both the phases use Discrete C-PSO for their operations, however the usage scenario varies considerably due to the varied requirements of both the phases. The below sections discusses the workflows in detail.

3.1 Service Selection:

User input is usually lexical and provides an abstract requirement. However, several independent components are required to complete the user's requirement. These components are identified and the services satisfying each of these components are identified by the Discrete Catfish PSO. Independent QoS requirements for each of the components are identified, which forms the fitness requirements. The major and basic requirement for the selection technique is to satisfy the fitness requirement of the user to the maximum possible extent. The service repository might not contain the service that can exactly satisfy the QoS requirements of the user. Hence this domain demands an optimal solution and not an accurate solution. This means that a solution that diverges slightly from the QoS requirement is acceptable. The divergence can be in the positive or in the negative magnitude. The process of service selection is presented in figure 1.

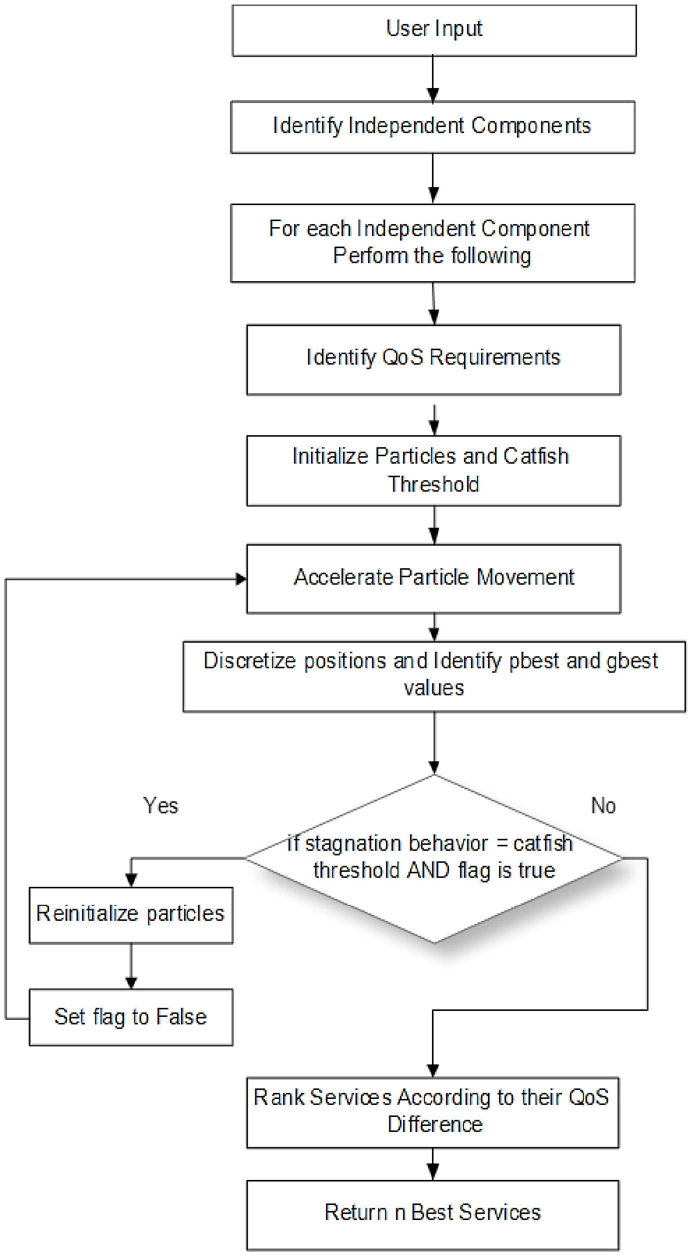

Figure 1: Service Selection Architecture

The search space is initialized with services and their QoS values are considered as the fitness values. The requirement is also added as a node in the search space and all the particles are distributed on the requirement node. Catfish threshold is initialized by the user. This determines the re-initialization level of the particles. Initial velocity of the particles is calculated using eq (1)

$$v_i \sim \cup \left(-\left|b_{up} - b_{lo}\right|, \left|b_{up} - b_{lo}\right|\right) \qquad (1)$$

Where Vi is the velocity bup and blo are the upper and lower bounds of the search space respectively.

Particles are accelerated in the search space. PSO operates on continuous space, hence the movement of particles are not defined between nodes. However, service selection is a discrete process, requiring a distinct service to be provided as the output. Hence PSO is modified to include the process of discretization as one of its components. Discretization is carried out after particle movement using eq (2).

$$p' = min\left(\sum_{j=1}^{n}\left(\sum_{k=1}^{d}\sqrt{\left(p_{ik} - n_{jk}\right)^2}\right)\forall i = 1top\right) \qquad (2)$$

Where Pik refers to the particle i's current location corresponding to dimension k, Njk refers to the kth dimension of node Ni.

At this point, every particle is distributed in the search space. pbest and gbest values are identified. Determination of the best solution for a particle is carried out using the fitness value of the solution. The fitness function used for the proposed work is

$$\text{fitness} = \frac{\delta_{QoS}}{\text{abs}\,(\delta_{QoS} - \rho_{QoS})} \qquad (3)$$

Where δQoS and ρQoS refers to the required QoS and the proposed QoS respectively. The major objective of this approach is to maximize the fitness function, which will ultimately provide the best service to suit the user's requirements. Velocity of particles are updated using eq (4)

$$v_{i,d} \leftarrow \omega v_{i,d} + \varphi_p \varphi_p\left(p_{i,d} - x_{i,d}\right) + \varphi_p r_p\left(g_d - x_{i,d}\right) \qquad (4)$$

Where rp and rg are the random numbers, Pi,d and gd are the parameter best and the global best values, Xi,d is the value current particle position, and the parameters ω, φp, and φg are selected by the practitioner.

The process is repeated until stagnation behavior reaches the catfish threshold. At this point, the particles have a huge probability of settling down in the local optima. Hence the particles are redistributed into the search space in a random manner and the process is repeated again. Due to the presence of redistribution mechanism, the particles are definitely moved out of the local optima, hence providing a near optimal solution.

The resultant services are ranked in accordance with their fitness values. A user defined threshold (n) is obtained, that defines the number of services to be shortlisted for the service orchestration module. Rank based shortlisting is carried out and the final n best services are passed to the service orchestration phase.

3.2 Service Orchestration:

Service orchestration phase performs the process of creating a service workflow by connecting the services selected in the previous phase. The initial step in performing service orchestration is to segregate the selected services according to their levels. Hence the service levels are equivalent to their corresponding workflow level. Search space for C-PSO is created using the selected services. The major difference between the service selection phase and the service orchestration phase is that the service selection phase identifies a single service as the solution, while the service orchestration phase needs to create a chain of services to create a workflow. The service orchestration architecture is presented in figure 2. Hence the fitness of the resultant workflow is identified by the aggregation of all the fitness values of the selected services.

The particles are initialized in services corresponding to the first level. Since several services are selected for each level, random initialization is performed. Catfish threshold is initialized and initial velocities are assigned to the particles. Particle movement is performed by adding the velocity component to the current position of the particle. Position of the particle is discretized to a service in the second level of the search space. This process is repeated until a complete workflow is obtained. Every particle identifies a single workflow. The QoS value corresponding to the workflow is identified and the pbest value is identified. The best workflow is identified and is labeled as gbest. This process of workflow identification is repeated until stagnation behavior. When the stagnation behavior reaches the catfish threshold, particle re-initialization is carried out. Several workflows are identified and they are ranked according to their fitness and n best workflows are filtered and passed to the user.

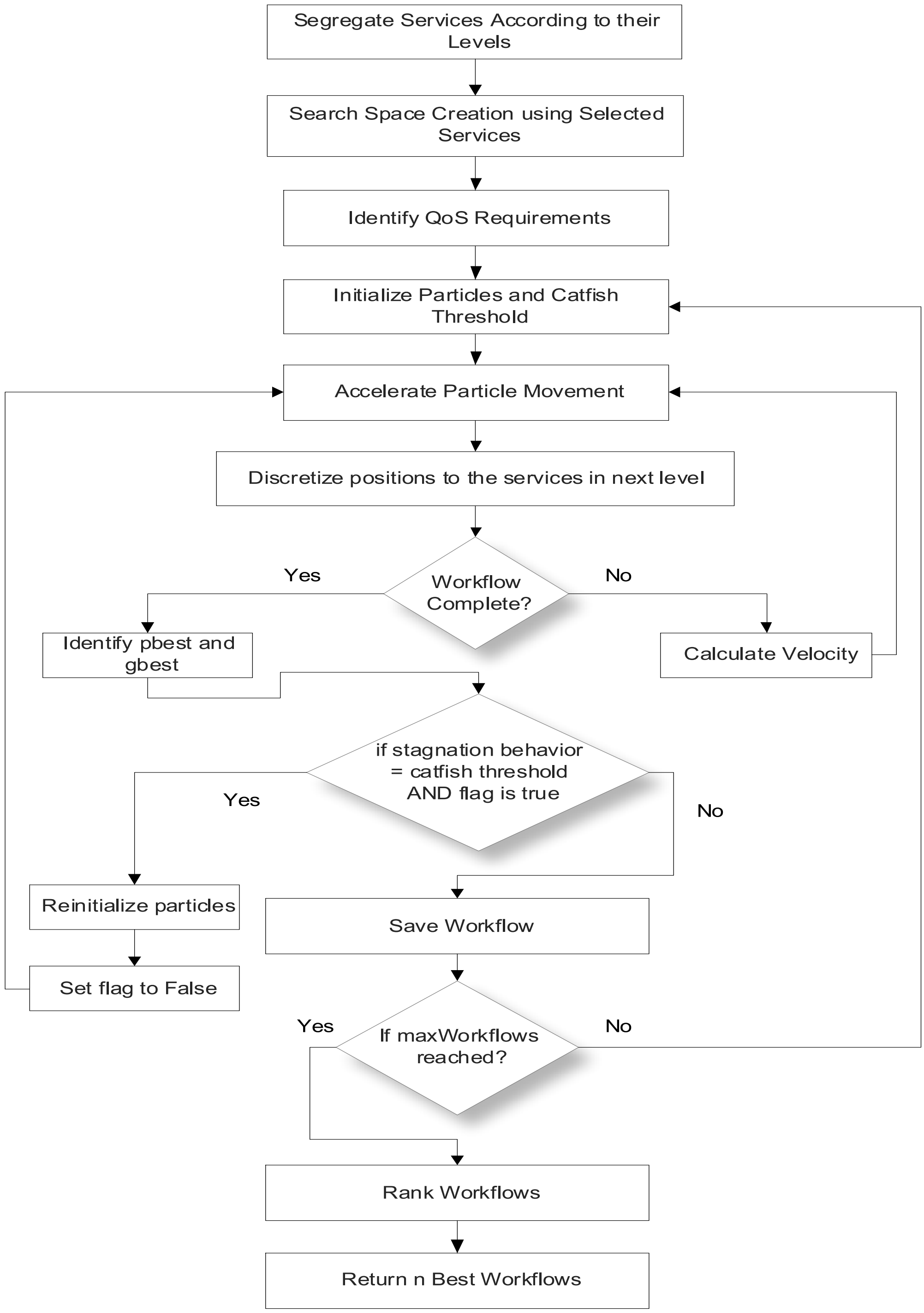

Figure 2: Service Orchestration Architecture

4. Results and Discussion:

Experiments were conducted by implementing C-PSO based service selection and C-PSO based service orchestration using C#.NET. Experiments were carried out using QWS Dataset 1.0 and QWS Dataset 2.0 [27, 28], containing 364 and 2503 instances and 13 and 10 attributes respectively. The QoS parameters considered are response time, availability, throughput, successability, reliability, compliance, best practices, latency, documentation, relevancy function and class level, while service name and its WSDL location are used for service identification. A comparison is carried out between the previous work (DPSOSA) of the authors with the current work (C-PSO). DPSOSA was designed specifically to reduce the probability of the system getting into local optima. The results are divided into two sections. The first section presents result from the service selection phase and the next section presents results from the service orchestration phase.

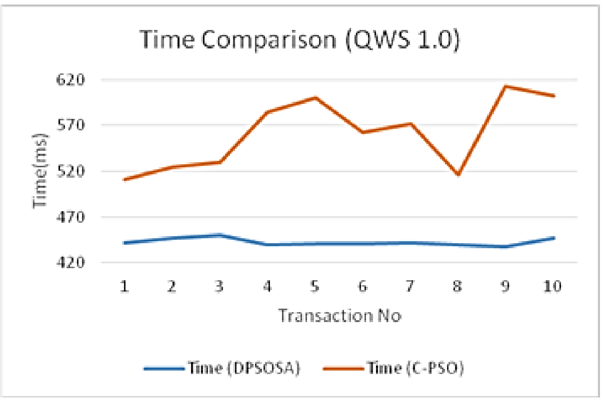

Figure 3: Time Comparison (QWS 1.0)

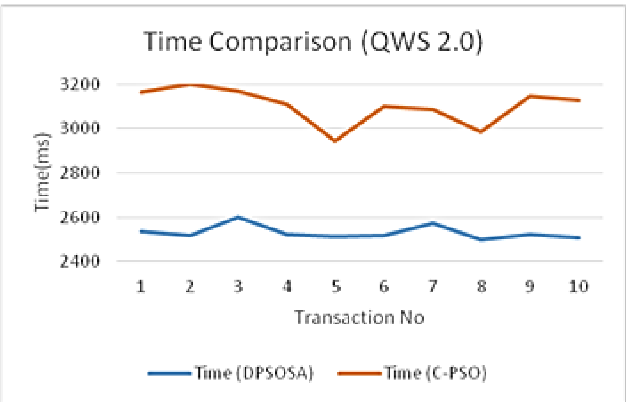

Figure 4: Time Comparison (QWS 2.0)

A time comparison between DPSOSA and C-PSO for QWS 1.0 and QWS 2.0 are presented in figures (3, 4). It could be observed that the time taken for service selection phase of C-PSO is higher than DPSOSA. This is attributed to the presence of the catfish particles in the selection algorithm. The catfish particles force the selection algorithm to operate at least twice even if the results are free from local optima. However, on close examination, it could be identified that the average time difference in QWS 1.0 is 130ms, while that of QWS 2.0 is 550ms. Both the time differences are observed to be quite small with a delay of < 1s. This is considerable in any real time system, hence the time delay becomes a factor with very low impact.

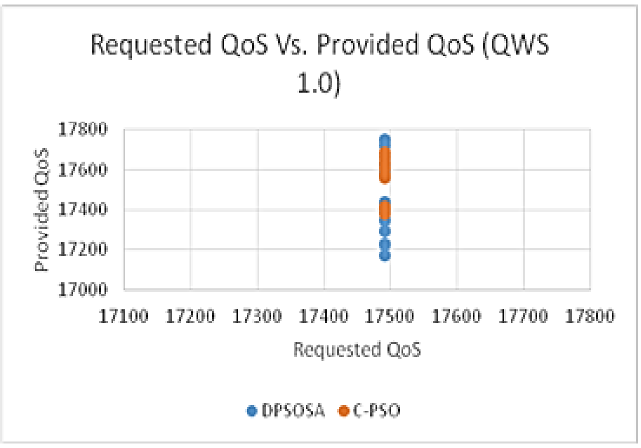

Figure 5: Requested QoS Vs. Provided QoS (QWS 1.0)

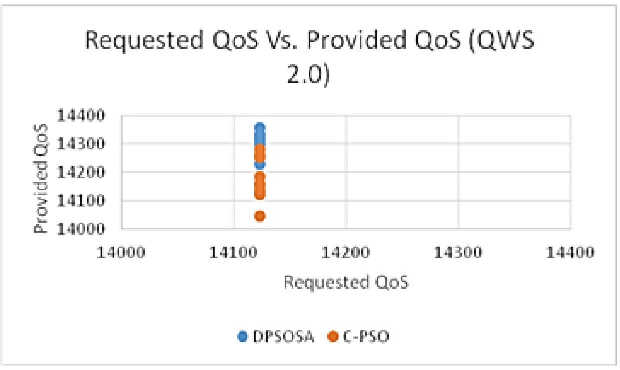

Figure 6: Requested QoS Vs. Provided QoS (QWS 2.0)

A comparison between the required QoS and the provided QoS of DPSOSA and C-PSO are presented in figures (5, 6). The required QoS of QWS 1.0 is set to 17492 and the required QoS of QWS 2.0 is set to 14123. The predictions closest to the required QoS both in the positive and negative directions are considered to be the best. Further priorities are provided to the predictions that fall above the requirements. It could be observed from both the figures that C-PSO exhibits closer prediction compared to DPSOSA. Hence it could be concluded that C-PSO exhibits higher efficiency in both allocating services.

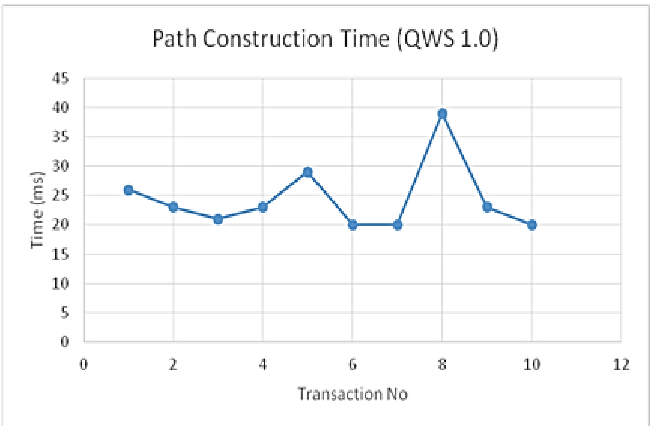

Figure 7: Path Construction Time (QWS 1.0)

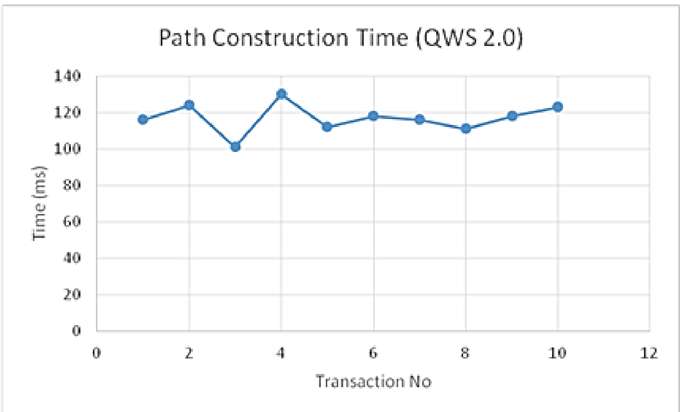

Figure 8: Path Construction Time (QWS 2.0)

Time taken for path construction using QWS 1.0 and QWS 2.0 are presented in figures (7, 8). Ten different requirements were provided to the system and the time taken for constructing paths is recorded. It could be observed that the maximum path construction time of QWS 1.0 is 40ms, while that of QWS 2.0 is 130ms. The average time for path construction was observed to be 25ms and 110ms respectively for QWS 1.0

and QWS 2.0. The overall time requirements come down to <1s, making the system an effective candidate for the service selection and the service orchestration process.

5. Conclusion:

Web service selection and service orchestrations are the major requirements of the current systems. This is due to the very compelling nature of ease of use provided by the commercial off-the-shelve components. However, issues arise during the process of selecting services. A huge number of services are available to solve a particular issue, each with its own set of QoS parameters. Selecting the optimal service is the current requirement. This paper presents an effective solution for service selection and service orchestration using a modified form of PSO called the Catfish-PSO. This algorithm is modified to provide discrete solutions, service selection and service orchestration. Experiments conducted on the proposed technique shows high levels of efficiency in the process of selection and orchestration.

6. References:

1. Zeng, L., Benatallah, B., Dumas, M., Kalagnanam, J., Sheng, Q., 2003, “Quality-driven Web services composition.” Proc. the 12th International Conference on the World Wide Web, pp.411–421.
2. Zeng, L., Benatallah, B., Ngu, A.H.H., Dumas, M., Kalagnanam, J., Chang, H., 2004, “QoS-aware middleware for Web services composition,” vol. 30, no.5. IEEE Transaction on Software Engineering, IEEE Computer Society, pp. 311–327.
3. Canfora, G., Penta, M.D., Esposito, R., Villani, M.L., 2005, “An approach for QoS-aware service composition based on genetic algorithms.” Proceedings of the 7th annual conference on Genetic and evolutionary computation. ACM, pp.1069–1075
4. Alrifai, M., Risse, T., 2009, “Combining global optimization with local selection for efficient QoS-aware service composition.” Proc. The 18th International Conference on theWorldWideWeb, pp.881–890.
5. Alrifai, M., Skoutas, D., Risse, T., 2010, “Selecting skyline services for QoS-based Web service composition.” Proc. the 19th International Conference on the World Wide Web, pp.11–20.
6. Wang, S., Hsu, C. H., Liang, Z., Sun, Q., & Yang, F., 2014, “Multi-user web service selection based on multi-QoS prediction.” Information Systems Frontiers, 16(1), pp. 143-152.
7. Shahand, S., Turner, S. J., Cai, W., & Khademi, H., 2010, “DynaSched: A dynamicWeb service scheduling and deployment framework for data intensive Grid workflows.” Procedia Computer Science, 1(1), pp. 593–602.
8. Dyachuk, D., & Deters, R., 2006, “Scheduling of composite web services.” In OTM Confederated International Conferences "On the Move to Meaningful Internet Systems" (pp. 19-20). Springer Berlin Heidelberg.
9. Chouiref, Z., Belkhir, A., Benouaret, K., & Hadjali, A., 2016, “A fuzzy framework for efficient user-centric Web service selection.” Applied Soft Computing, 41, pp. 51-65.
10. De Renzis, A., Garriga, M., Flores, A., Cechich, A., & Zunino, A., 2016, “Case-based reasoning for web service discovery and selection.” Electronic Notes in Theoretical Computer Science, 321, pp. 89-112.
11. De Renzis. A., Garriga, M., Flores, A., Cechich, A., and Zunino, A., 2014, “Semantic-structural assessment scheme for integrability in service-oriented applications.” In Computing Conference (CLEI), XL Latin American, pp. 1–11
12. Finlayson, M. A., 2014, “Java libraries for accessing the princeton wordnet: Comparison and evaluation.” In Proceedings of the 7th Global Wordnet Conference, Tartu, Estonia.
13. Mousa, A., & Bentahar, J., 2016, “An Efficient QoS-aware Web Services Selection Using Social Spider Algorithm.” Procedia Computer Science, 94, pp. 176-182.
14. Hai, Y., Yan, L., & Lu, G., 2012, “Dynamic Prediction QoS-Aware Web Service Composition Model.” International Journal of Digital Content Technology & its Applications, 6(12).
15. He, J., Chen, L., Wang, X., & Li, Y., 2013, “Web service composition optimization based on improved artificial bee colony algorithm.” Journal of Networks, 8(9), pp. 2143-2149.
16. Zhang, T., 2013, “QoS-aware Web Service Selection based on Particle Swarm Optimization.” JNW 2013, 9(3) pp. 565-570.
17. Temglit, N., Chibani, A., Djouani, K., & Nacer, M. A., 2016, “Distributed Approach for QoS Service Selection in Web of Objects.” Procedia Computer Science, 83, pp. 1170-1175.
18. Zeng, L., Benatallah, B., 2004, “Qos-aware middleware for web services composition,” IEEE Transactions on Software Engineering 30 (5) pp. 311–327.
19. Gao, C., Cai, M., & Chen, H., 2007, “QoS-aware service composition based on tree-coded genetic algorithm.” In Computer Software and Applications Conference, 2007. COMPSAC 2007. 31st Annual International Vol. 1, pp. 361-367

20. Vanrompay, P., 2008, “Genetic algorithm-based optimization of service composition and deployment,” in: SIPE ’08: Proceedings of the 3rd international workshop on Services integration in pervasive environments, New York, NY, USA, 2008, pp. 13–18.
21. Wu, M., Xiong, X., Ying, J., Jin, C., & Yu, C., 2011, “QoS-driven global optimization approach for large-scale web services composition.” Journal of Computers, 6(7), pp. 1452-1460.
22. Sasikaladevi, N., Arockiam, L., 2012, “Genetic approach for service selection problem in composite web service,” International Journal of Computer Applications 44 (4) pp. 22–29
23. Fanjianga, Y., Syub Y., Kuoc, J., 2016, “Search based approach to forecasting QoS attributes of web services using genetic programming”, Information and Software Technology, Vol 80, pp. 158–174
24. Xu, Y., Yin, J., Deng, S., Xiong, N. N., & Huang, J., 2016, “Context-aware QoS prediction for web service recommendation and selection.” Expert Systems with Applications, 53, pp. 75-86.
25. Wang, H. C., Chiu, W. P., & Wu, S. C., 2015, “QoS-driven selection of web service considering group preference.” Computer Networks, 93, pp. 111-124.
26. Iordache, R., & Moldoveanu, F., 2014, “QoS-aware web service semantic selection based on preferences.” Procedia Engineering, 69, pp. 1152-1161.
27. Al-Masri, E., and Mahmoud, Q.H., 2007, “Discovering the best web service,” (poster) 16th International Conference on World Wide Web, pp. 1257-1258.
28. Al-Masri, E. and Mahmoud, Q.H., 2007, “QoS-based Discovery and Ranking of Web Services,” IEEE 16th International Conference on Computer Communications and Networks (ICCCN), pp. 529-534.