

QOS Based Web Service Composition for Selection Using Improved Gravitational Search Algorithm

¹N. Hema Priya and ²S. Chandramathi

¹Department of Information Technology, PSG College of Technology,
Coimbatore, Tamil Nadu, India

²Department of Electronics and Communication Engineering,
SNS College of Technology, Coimbatore, Tamil Nadu, India

Abstract: With the drastic increase in the availability of functionally similar web services, the users take the herculean task of selecting the apt service that suits his needs. The services are although functionally similar, they differ with respect to their non functional parameters, in the name of Quality of Service (QoS). But still, dynamic optimal selection of services can be made possible by getting user preferences and by the use of computational intelligence techniques. QoS are analyzed and it is designed at the middleware to implement web service selection and composition. In this study, Improved Gravitational Search Algorithm (IGSA) is proposed to solve dynamic consumer needs and various QoS parameters are considered for composition of services. The results indicate that the proposed method shows improvement when compared with other methodologies such as Hybrid Intelligent Water Drops (HIWD) algorithm and Gravitational Search Algorithm (GSA).

Key words: Web service selection, composition, QoS, optimal, non functional parameters, IGSA, HIWD

INTRODUCTION

Service-Oriented Architecture (SOA) is an approach used to create an architecture based on the use of services. Some basic analytics are carried out by the services, other than validation and data production. SOA has many applications but one predominant implementation of SOA is web services. Web service is an application component, accessible over open protocols. Normal web service supports interoperable machine-to-machine interaction over the internet (Jaeger *et al.*, 2005). Communication is possible through Web services and it communicates with each other in a platform independent manner (Sharifi *et al.*, 2009). 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 to describe operation to execute or to exchange data with another Web service. The principal building blocks needed for today's Web Services Interaction are SOAP, UDDI and WSDL. For interaction, first the services are registered in UDDI repository and made available for invocation. The UDDI repository contains all the necessary information to identify Web Services along

with a URL that points to its corresponding WSDL file. The descriptions about the operations are available in the WSDL file. Secondly, service requestor sends query to the UDDI repository and finds the Web Services best suited for his need and in the third step, the WSDL file of that service is responded by the repository.

QoS requirements may change dynamically or failure of some services can occur at run-time which means that a quick response to adapt to the requests is important in web service selection. So the need of an effective and efficient service selection approach will increase (Jaeger *et al.*, 2005). Web services provided by different vendors have unpredictable characteristics. So, to select the highly relevant web service, QoS for web services are considered (Yu and Lin, 2005; Yang *et al.*, 2010). Web services having unpredictable nature so it is not an easy task to evaluate the desired QoS. To provide users a better service, it is necessary to identify the users' needs and QoS metrics for web services and based on that selection of available services.

Literature review: Non-functional qualities play an important role in all service related tasks, especially in discovery, selection and substitution of services. It is

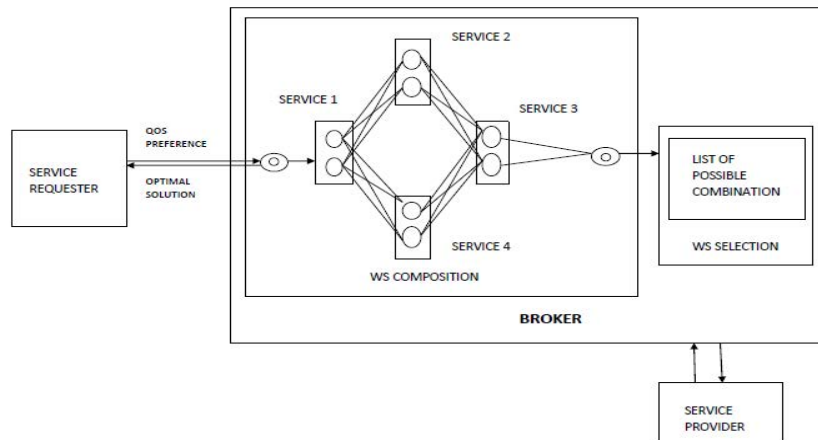


Fig. 1: System design

simple to imagine a scenario in which multiple services as shown in Fig. 1. In this case, the ability of the user to categorise between the services depends upon their non-functional qualities. To satisfy the user's requirements in better manner, it is first necessary to identify the users' needs and then identify all the possible QoS metrics for web services and based on it select the available service. Also, it is harder to expect the customers to consider all these QoS and specify their requirement. So, availability, reliability, response time, reputation and cost are the quality aspects considered for the proposed system which helps users to identify the highly optimized service among multiple services.

Jaeger *et al.* (2005) proposed the selection mechanism which defines a Simple Additive Weighting (SAW) method to rank the functionally similar Web services based on the user's QoS requirements. The SAW method discovers the score for web services all the way through summation of normalized QoS values which are multiplied by QoS preferences (weight). This service selection mechanism tries hard not to filter the functionally similar Web services. Moreover, based on the QoS requirements as the requester, this method does not provide desired (expected) QoS values for selection. Homa Movahednejad (Sharifi *et al.*, 2009) proposed a method to prove applicability and correctness of web service composition. It demonstrates how a web service composition approach addresses QoS aspects of customers. This method may involve in evaluating the non functional QoS parameters as trust and security.

Yu and lin (2005) proposed algorithms to maximize the user defined utility function value while meeting the end-to-end delay constraint and two optimal approaches to the service selection problem: the combinatorial approach, (Zhang, 2014) by modeling the problem as

the Multiple Choice Knapsack Problem (MCKP) and the graph approach, by modeling the problem as the constrained shortest path problem in the graph theory. Yang *et al.* (2010) presents a dynamic web services composition algorithm based on the optimal path method of the weighted directed acyclic graph. They employ web service dependence and user profile based service selection constraint calculation model, however, the model has a low efficiency under the dynamic real-time environment with a large number of web services. Genetic algorithm and immune algorithm provide good models to solve the difficult combinatorial optimization problems and have good performances in web services composition (Liu *et al.*, 2008).

Kritikos and Plexousakis (2007) describes about the basic web service discovery and deals with the requirements for web service discovery. The process involved are the matchmaking which find the list of relevant services in the registry and then selection services based on ranking approach. The advantage of such process are to select and execute the service to fulfill the customer expectation and also to Select the service satisfying the network parameters like availability, reliability, security, Integrity. But the service has different set of qualities, so considering all the parameters together have to be considered while selecting a service which leads to need for selection of composition based QoS aware Web service.

Bashir *et al.* (2010) Proposed an efficient dynamic web services discovery mechanism that can locate relevant and updated web services from service registries and repositories with times tamp based on indexing value and categorization for faster and efficient discovery of service. The proposed prototype focuses on quality of service issues and

introduces local cache concepts, CSP (constraint satisfaction problem) solver, aging and usage of translator.

Yu *et al.* (2007) selects the service with maximum utility function with QoS constraints. The algorithm used are combination model and graph based model. The combination model is based on the multi dimension multi choice 0-1 knapsack problem. In which services are represented in form of DAG. Moreover they are simplified models with high complexity. To improve there is need for other approaches like multi attribute optimization, constraint satisfaction problem, Genetic algorithm, Fuzzy techniques.

Zibanezhad *et al.* (2011) proposed gravitational search algorithm which is derived from concept of natural masses is determine the best web services combination. Simulation of final results showed that when comparing GSA with PSO algorithm, GSA has a significant potential for finding the ideal user combination in short time and with less memory is used for experiment. Liu *et al.* (2008) proposed user-aware Quality of Service (QoS) based Web services composition model is proposed. Under such model, a Web services selection method based on quantum genetic algorithm is proposed. Thus the algorithm uses quantum bit encoding and dynamically varies step-length quantum gate angle adjustment, the services are searched in neighbours and uses the dynamic punishment strategy to expand search scope and speed up convergence.

Tong and Zhang (2006) proposed a fuzzy multi attribute decision making algorithm for Web services selection based on quality of service is proposed which can select the most appropriate one with the highest degree of membership belonging to the positive ideal solution. But this process is time consuming and more cost. Fuzzy service selection constraints were overcome by make use of non-functional parameters. These parameters are manipulated in QoS manipulator and the resultant value is passed to fuzzy engine for finding the degree using min-max fuzzy rule based approach.

Spichakova (2013) proposed Gravitationally inspired Algorithm (GSA) and binary GSA heuristic algorithm for moore machines. They constructed system of N objects with position vector values and it consider objects with heavier mass attracts better when compare with objects with lower Mass. First initial masses are created randomly, next objects are evaluated. Secondly, the position of objects is changed based on forces acting on them and updated positions are calculated. This process continues until the best solution is achieved. The key difference between GSA and BGSA is each dimension of objects has only two values and the updating law of velocity has to

be changed only between these two values. They applied these methods for mealy machines too.

MATERIALS AND METHODS

Proposed model: In the proposed system, the solution to optimization problem is given through Improved Gravitational Search Algorithm (IGSA). This approach maximizes user satisfaction in selecting the web service by composing the qualities of web service and also addressing the multi criteria decisions and customer vague perception using IGSA. This has a number of different features such as use of various parameters and effective memory usage. The proposed methodology is focused to solve combinatorial optimization problem and is easily understandable. This method allows imprecise information as input, works by standardizing the QoS input values and find optimal composition paths among lots of possible paths. This method is experimented and the results are compared with other optimization techniques such as Gravitational Search Algorithm (GSA) (Kumar and Sahoo, 2014; Rashedi *et al.*, 2009) and HIWD. The results indicate that IGSA provides an optimal solution to the dynamically changing optimization problem. At the end, results are given with graphical representation and customer rating for selected service is updated. The overall procedure for the proposed system is as follows:

- Step 1: End user gives service request to the broker
- Step 2: Standardization of parameters in broker.
- Step 3: Broker analyze the given request and collect the list of services from the regist.
- Step 4: composition takes place between different candidate services
- Step 5: Possible list of combination will be given to the input of web service selection
- Step 6: After implementation of Improved GSA, the list of resulted service given to end user
- Step 7: Review given by user for selected service will be updated
- Step 8: QoS preferences will be given to Hybrid Intelligent Water Drops algorithm
- Step 9: Compare the results of Hybrid Intelligent Water Drops Algorithm, GSA and IGSA

Improved gravitational search algorithm: The algorithm is organized so that, there is a system of M no of objects, each objects described by a real position value. Improved GSA (IGSA) is based on law of gravity, it defines how objects are move depends on forces. Position of each vector can be as:

$$X_j = (x_{j1}, x_{j2}, \dots, x_{jm}), \quad m \in [1 \dots n]$$

- All these objects attract each other by a gravity force and this force causes a movement of all objects globally towards the objects with heavier masses
- The heavy masses correspond to good solutions of the problem
- The position of the agent corresponds to a solution of the problem and its mass is determined using a fitness function

By lapse of time, masses are attracted by the heaviest mass and this mass would present an optimum solution in the search space. The GSA could be considered as an isolated system of masses. It is like a small artificial world of masses obeying the Newtonian laws of gravitation and motion.

QoS attribute standardization: Different QoS values have different range of values, so it's not proper way to consider all values directly, so it's better to standardize them before use, for computing. QoS attributes can be divided into two categories positive attributes and negative attributes. For example availability, reliability, reputations are positive attributes as well as cost and response time values are considered as negative attributes. To standardize the positive and negative attributes following formulae can be used where I and k are considered as service instances.

For positive attributes:

$$Q_i K = Q_i K - \alpha K / \sigma K; \quad 1 \leq K \leq n$$

For negative attributes:

$$Q_i K = 1 - Q_i K - \alpha K / \sigma K; \quad 1 \leq K \leq n$$

IGSA mass value calculation: Objects mass values are computed based on the QoS quality measure as follows:

$$M_j(t) = \frac{m_i(t)}{\sum_k m_k(t)}$$

$$m_i(t) = \frac{\text{fitness}(t) - \text{worstval}(t)}{\text{bestval}(t) - \text{worstval}(t)}$$

where, m_j , m_i and m_k are mass values. $\text{Bestval}(t)$ and $\text{worstval}(t)$ can be derived from the following equation:

$$\begin{aligned} \text{Bestval}(t) &= \text{Maxfitness}(t) \\ \text{Worstval}(t) &= \text{Minfitness}(t) \end{aligned}$$

Fitness value calculation: The fitness value of each objects are calculated using the nonfunctional parameters as availability, reliability, response time, cost and reputation:

$$\text{fitness}(t) = \frac{\text{Reputation} + \text{Availability} + \text{Reliability} + \text{Response time} + \text{Cost}}{\text{time}}$$

IGSA force value calculation and updation: At a specific time t are compute the force that is applied to the one object with mass M_i by some object with mass M_j using following equation:

$$F_{jk}(t) = \frac{G(t)M_i(t)M_j(t)}{R_{jk} + \epsilon} (x_j - x_k)$$

where, ϵ constant value between 0-1. $G(t)$ is gravitational constant. R_{jk} is the Euclidean distance between two objects that distance can be calculated using given below equation:

$$R_{jk} = \|x_j(t) - x_k(t)\|$$

The re computation values of the velocity, acceleration and force will be updated for next iteration.

RESULTS AND DISCUSSION

Selection matrix evaluation: To select the service from multiple combinations, matrix multiplication among candidate services used. Based on the user QoS preferences the service is chosen from matrix multiplication result. Figure 2 shows the possible combinations in proposed architecture model. Consider service S_i has $S_{i1}, S_{i2}, \dots, S_{in}$ set of candidate services:

$$\begin{Bmatrix} \begin{bmatrix} QoS - S_{11} \\ QoS - S_{12} \\ QoS - S_{13} \\ QoS - S_{14} \\ QoS - S_{15} \end{bmatrix} \begin{bmatrix} QoS - S_{22} \\ QoS - S_{23} \\ QoS - S_{24} \\ QoS - S_{25} \end{bmatrix} \begin{bmatrix} QoS - S_{31} \\ QoS - S_{32} \\ QoS - S_{33} \\ QoS - S_{34} \\ QoS - S_{35} \end{bmatrix} \end{Bmatrix}$$

Here, ($S_{11}, S_{12}, S_{15}, S_{21}, \dots, S_{25}, \dots, S_{31}, \dots, S_{35}$) are candidate services of services in middleware.

Hybrid intelligent water drops algorithm: Hybrid Intelligent Water Drops algorithm (HIWD) is a swarm-based optimization algorithm which has been

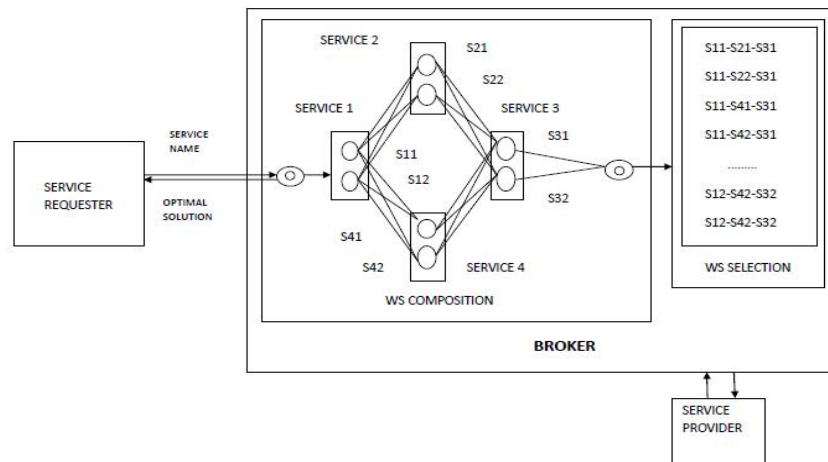


Fig. 2: Example for list of all possible combination without algorithm

inspired feasible path among 'n' no of alternative paths. These feasible path finding is based on river system and the actions and reaction that take place between water drops in the river. Velocity is considered as the main feature in natural river. It assumes each water drop of a river can carries an amount of soil. Based on velocity, soil can be removed from one place to another and this deeper part can have more soil. An intelligent water drop will always try to choose an easy path rather than the harder path among multiple paths. These ideas are embedded into the web service selection and composition for providing efficient QoS web services. The important properties of water drops are considered as follows:

- More soil is gathered by high speed water than low speed water
- The velocity of a water drop increases more on a path with low soil than a path with high soil
- Path with less soil is preferred by water drops than the path with more soil

In an environment, there is more than one path available from source to destination and the position of the destination may be known or unknown. The goal is to find the optimum solution or path from the source to the destination.

Hybrid IWD-Initialization of values: At first, the parameter values (availability, reliability and response time) are initialized and fitness value is calculated, then the fitness value is calculated for each best individuals. Next Mutation and crossover are used to produce the new population of the best individuals. Again the fitness value of new population will be calculated and this value is considered as initial value for objects. The graph (N, E) is set for the problem. The quality of the total-best solution

BTB is initially set to the worst value: $q(BTB) = -\infty$. The maximum number of iterations (iterationmax) given by the user. The iteration count iteration count is set to zero. The number of water drops NOIWD is set to a positive integer value. The velocity is updated using the following parameters, $xv = 1$, $yv = 0.01$ and $zv = 1$. For soil updating, $xs = 1$, $ys = 0.01$ and $zs = 1$. The local soil updating parameter ρ_{lon} which is a small positive number less than one is set as $\rho_{lon} = 0.9$. The ρ_{lwd} is for global soil updating which is chosen from $[0, 1]$, so, $\rho_{lwd} = 0.9$. Every IWD has a visited node list VIC (IWD) which is initially empty: $VIC(IWD) = \{\}$. Each IWD's velocity is set to InitialVelocity. Initially all IWDs are set to have 0 amount of soil.

Updation of soil, velocity and best solution: The path from node i to j traversed by that IWD carries some soil based on velocity acted on them:

$$\begin{aligned} \text{Soil value}(i, j) &= (1 - \rho_{lon}) \text{Soil value}(i, j) - \rho_{lon} \Delta \text{Soil value}(i, j) \\ \text{Soil value IWD} &= \text{Soil value IWD} + \text{Soil value}(i, j) \end{aligned}$$

where for the IWD moving on the path from node i to j , compute the soil $\Delta \text{soil value}(i, j)$ that the IWD loads from the path by:

$$\text{Soil value}(i, j) = \frac{xs}{ys + zs \cdot \text{time2}(i, j; \text{veloiwd})(t + 1)}$$

Then, Update the soils on the paths that form the current iteration-best solution BIB b:

$$\begin{aligned} \text{Soil value}(i, j) &= (1 + \rho_{lwd}) \cdot \\ &\text{Soil value}(i, j) - \rho_{lwd} / (\text{NOTB} - 1) \end{aligned}$$

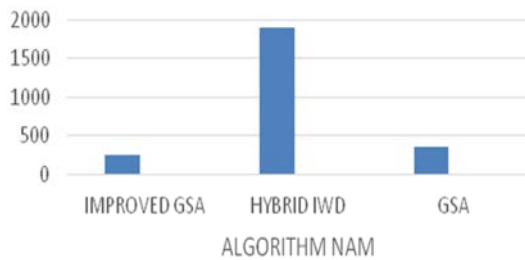


Fig. 3: Performance comparison chart

Update the total best solution BTB by the current iteration-best solution BIB using the following equation:

$$\text{BIB} = \begin{cases} \text{BIB} & \text{if } q(\text{BTB}) \geq q(\text{BIB}) \\ \text{BIB} & \text{otherwise} \end{cases}$$

The algorithm stops here with the total-best solution BTB. The total best solution used to select the best services.

Result analysis of IWD and AIS: When a web service is frequently used by the customers, one of the main reasons may be its quicker response. The provider gets more number of hits only when the reputation level is high. Using this technique, the proposed methods are analyzed. The results are compared with the help of computation time. The QWS dataset is used as input and the algorithm is made to run on NetBeans IDE. The experimental results show that IGSA takes 244 milliseconds for execution, HIWD takes 1896 milliseconds and GSA takes 360 milliseconds as execution time. IGSA identifies the difference between solutions and based on that it moves quickly towards the convergence point to select the best Web services combination among multiple compositions. Always the user would prefer to choose service which is returned immediately at once the query was posted and that is considered as the efficient method. Based on the experimental results obtained, the technique using IGSA is faster and efficient, when compared with HIWD and GSA algorithms. Figure 3 shows the comparison of methods using EA based algorithms.

CONCLUSION

With the increase in number of web services, it is necessary to solve the problem of service selection to meet the user's need in the web service composition. For meeting the users' requirements, QoS is a decisive factor for selecting the most suitable service from a set of candidate services with the identical functionality. Thus the approach about the computation of the QoS is also

becoming more important in the service selection. In order to solve these composition problems, IGSA and HIWD algorithms are proposed. To verify the best web service composition methods, above algorithms are implemented and compared with GSA. The experimental result shows that Improved Gravitational Search Algorithm's execution time is 244 milliseconds, Hybrid IWD algorithm's execution time is 1896 milliseconds and GSA takes 360 milliseconds for its execution. When comparing IGSA with the GSA and Hybrid IWD algorithm, IGSA gives significant potential for identifying the optimal user combination in the least time and therefore it's considered as good choice for practical implementation. This concludes service selection technique using IGSA is the best and indicates that the response time of IGSA is nearly 30% better than other evolutionary algorithms based techniques.

REFERENCES

- Bashir, S., F.H. Khan, M.Y. Javed, A. Khan and M.S.H. Khiyal, 2010. Indexer based dynamic web services discovery. *Int. J. Comput. Sci. Inf. Security*, 7: 153-159.
- Jaeger, M.C., G. Muhl and S. Golze, 2005. Qos-Aware Composition of Web Services: An Evaluation of Selection Algorithms. In: *On the Move to Meaningful Internet Systems 2005 CoopIS, DOA and ODBASE*, Robert, M. and T. Zahir (Eds.). Springer Berlin Heidelberg, Berlin, Germany, ISBN: 978-3-540-29736-9, pp: 646.
- Kritikos, K. and D. Plexousakis, 2007. Semantic QoS-based web service discovery algorithms. *Proceedings of the 5th European Conference on Web Services*, November 26-28, 2007, Halle, Germany, pp: 181-190.
- Kumar, Y. and G. Sahoo, 2014. A review on gravitational search algorithm and its applications to data clustering and classification. *Int. J. Intell. Syst. Appl.*, 6: 79-93.
- Liu, X., Y. Feng and J. Kerridge, 2008. Automated responsive web service evolution through generative aspect-oriented component adaptation. *Int. J. Comput. Appl. Technol.*, 31: 83-93.
- Rashedi, E., H. Nezamabadi-Pour and S. Saryazdi, 2009. GSA: A gravitational search algorithm. *Inform. Sci.*, 179: 2232-2248.
- Sharifi, M., H. Movahednejad, S.G.H. Tabatabaei and S. Ibrahim, 2009. An effective access control approach to support web service security. *Proceedings of the 11th International Conference on Information Integration and Web-based Applications and Services*, December 14-16, 2009, ACM, Kuala Lumpur, Malaysia, ISBN: 978-1-60558-660-1, pp: 529-535.

- Spichakova, M., 2013. An approach to the inference of finite state machines based on a gravitationally-inspired search algorithm. *Proc. Estonian Acad. Sci.*, 62: 39-46.
- Tong, H. and S. Zhang, 2006. A fuzzy multi-attribute decision making algorithm for web services selection based on QoS. *Proceedings of the 2006 IEEE Asia-Pacific Conference on Services Computing (APSCC'06)*, December 12-15, 2006, IEEE, New York.
- Yang, Z., C. Shang, Q. Liu and C. Zhao, 2010. A dynamic web services composition algorithm based on the combination of ant colony algorithm and genetic algorithm. *J. Comput. Inf. Syst.*, 6: 2617-2622.
- Yu, T. and K.J. Lin, 2005. Service selection algorithms for Web services with end-to-end QoS constraints. *Inform. Syst. e-Bus. Manage.*, 3: 103-126.
- Yu, T., Y. Zhang and K.J. Lin, 2007. Efficient algorithms for Web services selection with end-to-end QoS constraints. *ACM. Trans. Web (TWEB.)*, 1: 1-26.
- Zhang, T., 2014. QoS-aware web service selection based on particle swarm optimization. *J. Networks*, 9: 565-570.
- Zibanezhad, B., K. Zamanifar, R.S. Sadjady and Y. Rastegari, 2011. Applying gravitational search algorithm in the QoS-based Web service selection problem. *J. Zhejiang Univ. Sci. C.*, 12: 730-742.