I.J. Modern Education and Computer Science, **2015, 2, 56-62**
Published Online February 2015 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijmecs.2015.02.08

A GA-Tabu Based User Centric Approach for Discovering Optimal Qos Composition

Vivek Gaur
Birla Institute of Technology, Computer Science Department, Jaipur, 302017, India
Email: vivekgaur@bitmesra.ac.in

Praveen Dhyani
Banasthali University, Computer Science Department, Jaipur 302001, India
Email: dhyani_p@yahoo.com

O. P. Rishi
Kota Engineering College, Computer Science Department, Kota 324010, India
Email: omprakashrishi@yahoo.com

Abstract—Cloud computing is an emerging internet-based paradigm of rendering services on pay- as -per -use basis. Increasing growth of cloud service providers and services creates the need to provide a tool for retrieval of the high-quality optimal cloud services composition with relevance to the user priorities. Quality of Service rankings provides valuable information for making optimal cloud service selection from a set of functionally equivalent service candidates. To obtain weighted user-centric Quality of Service Composition, real-world invocations on the service candidates are usually required. To avoid the time-consuming and expensive real-world service invocations, this paper proposes framework for prediction of optimal composition of services requested by the user. Taking advantage of the past service usage experiences of the consumers more cost effective results are achieved. Our proposed framework enables the end user to determine the optimal service composition based on the input weight for individual service Quality of Service. The Genetic algorithm and basic Tabu search is applied for the user-centric Quality of Service ranking prediction and the optimal service composition. The experimental results proves that our approaches outperform other competing approaches**.**

Index Terms—Cloud services, service utility, QoS, genetic algorithm, Tabu search.

I. INTRODUCTION

As the Cloud Computing is gaining popularity, finding the optimal combination of services that are relevant to the user query becomes increasing necessity. The main reason is that the need for the particular type of service is user dependent. Each user has different levels of priority for different services available on the cloud specially with respect to the cost. Therefore, we need efficient tool for responding to the user cloud service requirement with QoS combination in a cost effective manner and with high relevance. Quality-of-Service (QoS) parameters as response time, throughput, price etc. are important factors to be considered for determining the optimal service combination. The required QoS values are well recorded in a formal document called Service Level Agreement (SLA) as a contract between the service provider and end-user [1-2].

The QoS aware service combination selection aims to determine the best combination of cloud services candidates that satisfies a set of user constraints as per given SLA. The selected service composition is the one which maximizes the user related utility function.

In case of multiple attribute type used for optimization the selection problem becomes a multidimensional optimization problem.

On the basis of the fact that the priority and weight-age for QoS attributes, such as price, response time etc. can vary with respect to end user demands, the task of prediction of optimal user-centric service combination has gained popularity. A number of service selection approaches have been proposed [1, 3-7]. However, none of the approach fully respond to all the aspects of the user needs. The service selection system can be viewed as a collection of cloud services provided by different service providers, a collection of user requested service and method for determining the utility degree of services with respect to the user demands.

Firstly, most available service selection approaches do not consider the individual service specific QoS priorities. QoS values for a service combination are usually determined by aggregated QoS values of the services provided by the various providers with different location and network. Since the priority of user for one type of service QoS parameter differs from the another, the selection method needs to enable the end user to dynamically assign the priorities for each service QoS value according to need and budget.

Secondly, most of the approaches emphasized more on the optimization of the selection methods to decrease the computation time and effective utility value of the service combination for the end user. Based on the fact that the actual performance may vary dynamically thus end user must have a choice to accept or reject the service combination determined by the selection method. Further the sample space of service selection must be dynamic and the combination once not accepted by user must not participate in subsequent selection invocations for the user.

Therefore, we propose a selection approach based on cloud model. Our approach mainly entails:-

a. We address the problem of cloud selection service and present the effect of user-centric QoS Parameters.
b. We propose a model called user-centric discovery of optimal QoS composition. To achieve this we adopt GA based cloud model for finding the optimal service composition. A basic Tabu method is applied to deal with local optima problem eliminating chances of offering same composition on subsequent invocations.
c. We demonstrate our approach experimentally with three type of services as a sample input and shows the performance improvement on every subsequent selection call by the user.

The rest of the paper is organized as follows. The section II introduces the background and related work for the service composition selection. Section III explains the proposed approach framework. Section IV presents the experiments and Section V concludes the paper.

II. Related Work

A number of approaches are suggested to address the problem of selection of web service composition by many researchers[3-7]. The work in [3] shows the dynamic attributes based provision of service publication, discovery and selection which expresses the exact state and characteristics of cloud services. In work [4] gives a new approach of optimal service selection with three considerations, optimization through loops peeling, negotiation of QoS parameters and group of global constraints. This approach lacks the consideration for elimination of redundant service candidates resulting in large search space with possibility of same service offered with different level of QoS by different providers. In [6], the algorithm are heuristic in nature and gives more efficient rather the best or exact composition. There are approaches based on the service QoS parameters, have been proposed for service selection [8-9], which identifies optimal service from a set of functionally similar candidate services. Other similar approaches which proposed the dynamic service selection [5], skyline service selection [1], heuristic service selection [7].The work [10, 11] suggested a ranking-oriented collaborative filtering approach to rank movies books in digital libraries. A hybrid GA-Tabu approach was presented in the work [13]. The work in [14] proposed meta-heuristic based approaches using Tabu and hybrid Genetic algorithm. The limitation of the use of Genetic Algorithm is the early maturity of the solution with no scope for further improvement. The use of Tabu search approach solves the optimization problem and is capable to get escape from trap of local optimality.

Tabu search is used to select the candidate service/ chromosome with the best fitness after the crossover and mutation operation by Genetic Algorithm. The approach increased the diversity of population and more search space with further scope of optimized solution.

Most of the approaches addressed the problem considering it as providing the optimized service with QoS parameter in general for the end user. Different from these approaches, our work proposes a model to provide the user-centric service composition by allowing the user to assign the priorities in form of weights for each QoS parameters dynamically.

III. Proposed Work

In our proposed approach, we present a user- centric blending of the genetic algorithm and Tabu search for optimal cloud service composition as shown in fig.1. Selection of service composition is performed considering the users weighted priorities for the QoS parameters.

A. BasicTabu Search

Tabu search is an improvisation of the basic local search method. The basic local search limitation is called the local optima which means the search is confined to a local search space leading to degradation of the potential solution quality.

Tabu search makes a use of memory consisting of a list of recent search results that will not participate in selection process for next given N number of searches. This ensures that every time when function is called the searching will be done by exploring new search space[18].

a. Quality of the solution depend on the fulfillment of the requirements and constraints according to the individual service preferences.
b. All the compulsory and optional constraint types are fetched to the method to evaluate the service quality.

Based on the individual's past experience of the related problem the method enables the dynamic tailoring to resolve and meet the specific constraints violations.

B. Basic Genetic Algorithm (BGA)

Genetic algorithm is based on Darwin's theory of evolution and follows an evolutionary process. In Genetic Algorithm the chromosomes represent the candidate solutions and a set of solution is called as Population. New solutions (offsprings) are generated from the randomly produced population according to their fitness values. Better solutions are expected to be produced on every subsequent search with higer fitness value[17].

Steps required for BGA are as follows:

a. Randomly generate an initial population.
b. Compute the fitness of all selected solutions.
c. While terminating condition is false.
d. Select the solutions with higher fitness values to

make new reproduction pool.
e. Crossover the selected solutions.
f. Mutate the chosen solutions.
g. Compute the fitness value after mutation.
h. Produce a new population.
i. End While

C. System Architecture

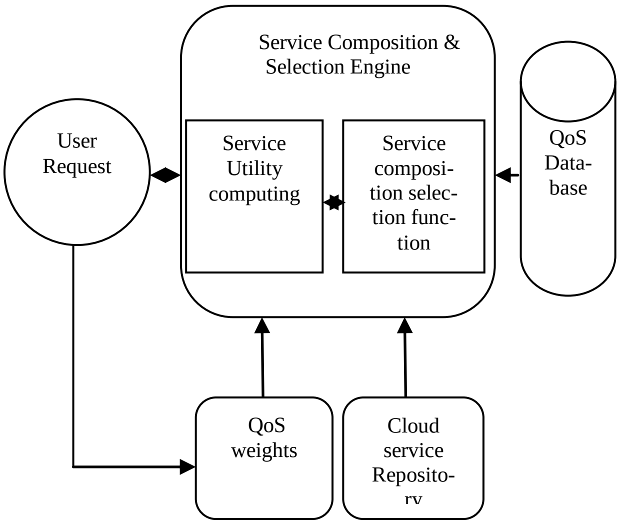
Service Composition & Selection Engine
User Request
Service Utility computing
Service composi-tion selec-tion func-tion
QoS Data-base
QoS weights
Cloud service Reposito-ry

Fig 1. System Architecture

- User Request - End User request for a composition of cloud services.
- QoS Weights - User specific priority values assigned for each QoS parameter.
- Cloud Service Repository - The Cloud service providers put all the functional and non-functional attributes of services in the repository.
- Service Composition & Selection Engine - determines the utility of possible service compositions and select the optimal one as per the weighted QoS requested by user.
- QoS Database **-** contains the updated information for QoS parameters considered by the service providers.

D. Service Composition Selection

Services with the identical functions and different QoS are escalating with the propagation of Service Oriented Architecture(SOA). It is required to identify and choose the best service composition from various schemas in order to meet the QoS requirement and optimize the satisfaction of users. The service composition is designed to compose the available services to generate a new value-added user centric service composition, which is one of the required features of web services.

This section presents the steps followed for selection of service composition with respect to user's QoS parameters. Section (a) presents the overall design of the selection algorithm. It comprises of the reading of user input weight for each QoS parameter, call to the GA_Tabu( ) function , checking with Tabu list and then present the result to the end user. Section (b) computes the fitness values of each candidate solution produced using GA_Tabu( ) function. Section (c) contains the definition of the GA_Tabu( ) function.

a. Design of User-Centric Optimal Service Composition

The optimal service selection approach presented in our work uses genetic algorithm in conjunction with concept of Tabu-Search. The two important factors have been considered in designing the service composition selection algorithm. Firstly, since every individual user has different requirements and priorities for different quality issues therefore the each end-user is enabled to input the weight for different QOS parameters according to its own priorities. Secondly, the chance of getting caught in local -optima is handled using the concept of Tabu-Search.

The selection process ranks the alternative compositions on the basis of QoS parameters and user specified weight to each QoS parameter. The compositions form chromosomes for the GA. A Tabu-List is maintained to keep track of earlier compositions rejected by particular classes of customers. The scheme of interaction is presented in Fig.2 and detailed algorithm is presented below.

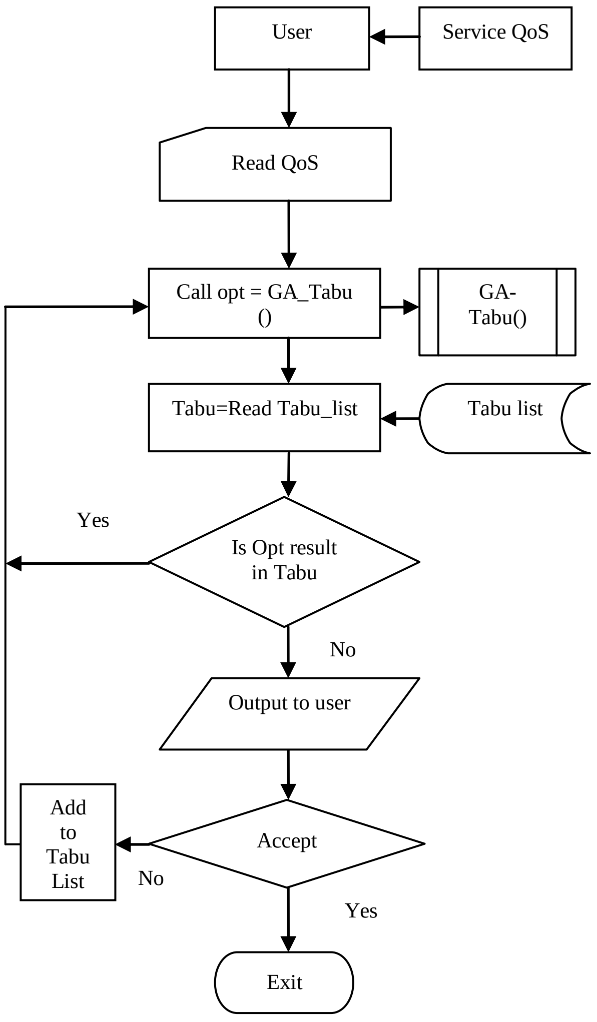

Fig 2. Flow Chart of Proposed model

b. Chromosome (composition) Fitness Computation

In the service composition, the service candidates have different values of the QoS parameters. The fitness function F(C) in (1) used for finding the utility of the composition is similar to the work by T. Yu, Y. Zhang and K.J. Lins [6] where the minimum and maximum aggregated values of the i^{th} QoS parameter of service composition C are computed as in (2) and (3).

$$F(C) = \sum_{j=1}^{r} \frac{Q_j^{max} - W_{j,i}\, q_j(C)}{Q_j^{max} - Q_j^{min}} \quad (1)$$

$$Q_j^{max} = \sum_{i=1}^{n} Q_{i,j}^{max},\ Q_{i,j}^{max} = \max_{\forall s_{ik} \in s_i} q_j\,(c_{ik}) \quad (2)$$

$$Q_j^{min} = \sum_{i=1}^{n} Q_{i,j}^{min},\ \ Q_{i,j}^{min} = \min_{\forall s_{ik} \in s_i} q_j\,(c_{ik}) \quad (3)$$

where, F(C) represents the overall utility of a composite service composition, i = 1 to n represents the service type (service1, service2, service3) and j = 1 to r represents the QoS parameters(price, response time, reliability and throughput) W_{ji} is the weight(0 to 1) assigned by the end user for j^{th} QoS of i^{th} service. $Q_{i,j}^{max}$ is the maximum value of the j^{th} parameter in all service candidates/ chromosomes of a i^{th} service category, similarly $Q_{i,j}^{min}$ is the minimum value. Q_j^{max} is the maximum value of the j^{th} parameter of a composite composition C and similarly Q_j^{min} is the minimum value. $q_j(c_{ik})$ is the j^{th} QoS parameter values for kth candidate of i^{th} service type.

The values of the QoS parameters for a service composition are aggregated by the selected chromosomes / service candidates using the aggregation functions presented in Table 1. In this work only sequential composition model is considered for aggregation similar to the work by Shangguang et.al [15]. The techniques presented in [16] can be used to transform the other models like parallel, conditional and loops to sequential model.

QoS parameters can be partitioned into two types: constructive and destructive QoS parameters. Constructive parameters mean that the higher the parameter value is, the better the quality is (e.g., reliability, Throughput). Destructive parameters means higher the parameter value is, the inferior the quality is (e.g., price, response time). So the aim to select the service composition which keeps the constructive parameters maximum and destructive parameters as minimum. There are other QoS parameters like availability and reputation also can be considered as in [15] to form a part of service composition selection. The inclusion of the parameters (availability and reputation) need the study of external factors like experience, previous performance history, SLA violation and market value of the service provider. In our work we only consider the four attributes (price, throughput, reliability, response time) in this paper for the sake of simplicity.

Table 1. QoS Aggregation Functions

QoS Attributes	Sequential Composition
Price	q(c) = $\sum_{i=1}^{n} q(c_i)$
Response Time	q(c) = $\sum_{i=1}^{n} q(c_i)$
Reliability	q(c) = $\frac{1}{n}\sum_{i=1}^{n} q(c_i)$
Throughput	q(c) = $\min_{i=1}^{n} q(c_i)$

c. *Function GA_Tabu()*

```
/*Input Service Candidates: (Options X 4): 4 relates to the QoS parameters namely: 1. Price 2. Response Time 3. Reputation 4. Throughput, Suppose there are 3 service-elements namely: 1.Storage 2.Software 3.Processing and Suppose no. of associates for each service element are: 1.Storage=10, 2. Software = 5 3.Processor: 4 */
/*TABU_LIST: List of compositions rejected ('Tabu') during Service Composition presentation to user; NULL on first execution.*/

/* Initialize GA*/
1.  Initialize 'Population' of size $N_P$- i.e. $N_P$ number of chromosomes each represented as a vector of length $N_S$ containing a combination of service_vendor from each service category.
2.  W=Get User's weight for each QoS parameter to use in fitness computation.
3.  Repeat Steps (3-6) until specified no. GA iterations
4.  Compute Fitness of each chromosome using (1) to compute Utility.
5.  Determine the Best Rank chromosomes to fill the reproduction pool of size ReprodPoolSize.
6.  Apply Single-random-point crossover over the chromosomes in the reproduction pool and generate chromosome-population of size $N_P$ for next iteration.
7.  Apply mutation operation over each chromosome in population checking as per given mutation probability.
8.  On completing the GA iterations BEST_COMP = Compute Fitness of each chromosome using Equation (1) and get the Best Rank chromosome.
        /* Now Apply Tabu*/
9.  If (Match_in(BEST_COMP, TABULIST))
10. Goto step 1
11. Else Present the BEST_COMP to user and ask for Selection/Rejection.
12. If Rejected
13. Enter BEST_COMP into TABULIST
14. Goto Step 1
15. Else
16. Display Success and Exit.
```

IV. Experiments

In this section, we present the results of experiments conducted using our proposed work. First we present the computation of optimal service composition and then the improvement is shown in every subsequent search request by the end user.

A. *Experiment Setup*

In this experiment, we uses a randomly generated dataset that contains the values of four QoS parameters, namely Price, Response Time, Throughput, Reliability.

The number of different service type is considered 3 (i.e. Storage, Software, Processing) and the number of service candidates per service type varies from 1 to 10 as shown in Table 2,3,4 respectively. All the experiments are conducted on a computer system with Intel(R) core(TM)i7-3632QM CPU @ 2.20GHz, 8.0 GB of RAM, Windows 8 and Matlab 7.6.0.

Table 2. Storage service

Service\ QoS	Price (rupees)	Response Time (seconds)	Throughput (0-1)	Reliability (0-1)
S1	6594.5	0.3	0.7	0.5
S2	2549.4	1.6	0.1	0.1
S3	5093.5	2.0	0.4	0.7
S4	8642.4	4.4	0.1	0.9
S5	6520.0	4.1	0.9	0.0
S6	0908.3	4.8	0.5	0.7
S7	6196.0	2.3	0.1	0.2
S8	2225.2	1.3	0.0	0.3
S9	8724.8	3.3	0.0	0.5
S10	3487.4	3.7	0.5	0.5

Table 3. Software Service

Service\ QoS	Price (rupees)	Response Time (seconds)	Throughput (0-1)	Reliability (0-1)
S1	2395.1	59	0.7	1.0
S2	1822.8	55	0.2	0.6
S3	2937.4	79	0.7	0.5
S4	3472.2	11	0.3	0.9
S5	0578.4	61	0.2	0.8

Table 4. Processing Service

Service\ QoS	Price (rupees)	Response Time (seconds)	Throughput (0-1)	Reliability (0-1)
S1	0341.6	9.3	0.0	0.0
S2	1414.1	1.7	0.0	0.1
S3	1117.6	8.7	0.0	0.1
S4	1663.1	0.2	0.0	0.1

B. *Experiment Analysis*

The experiment conducted on the input QoS values for three services (service1: storage, service2: software, service3: processing) as shown in fig.3. The output service composition 6, 4, 1 as shown in fig.4 represents the composition of 6th storage service, 4th software service and 1st processing service having the highest utility value with respect to the user input QoS values for different service types. The plot presented in fig.5 the shows improvement in service composition utility values on subsequent iterations. In case the user rejects (Tabu) the output composition (6, 4, 1) then the result is added to the Tabu-list and new composition is determined excluding the current result. The fig.6 and fig.7 shows the new service composition (8, 2, 4) and utility-iteration plot respectively.

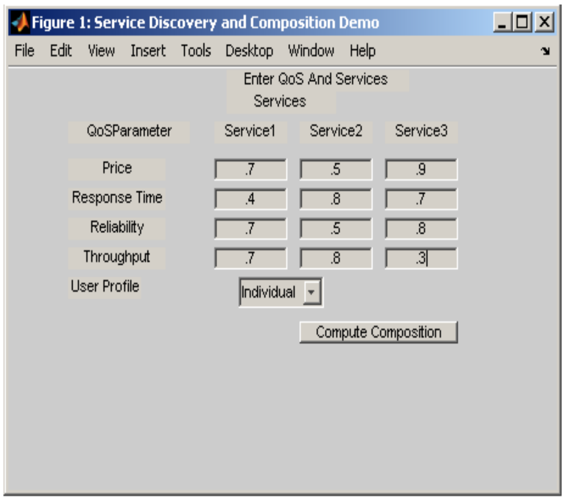
Figure 1: Service Discovery and Composition Demo
File Edit View Insert Tools Desktop Window Help
Compute Composition

Enter QoS And Services Services

QoSParameter	Service1	Service2	Service3
Price	.7	.5	.9
Response Time	.4	.8	.7
Reliability	.7	.5	.8
Throughput	.7	.8	.3

User Profile: Individual

Fig 3. Service QoS values input screen

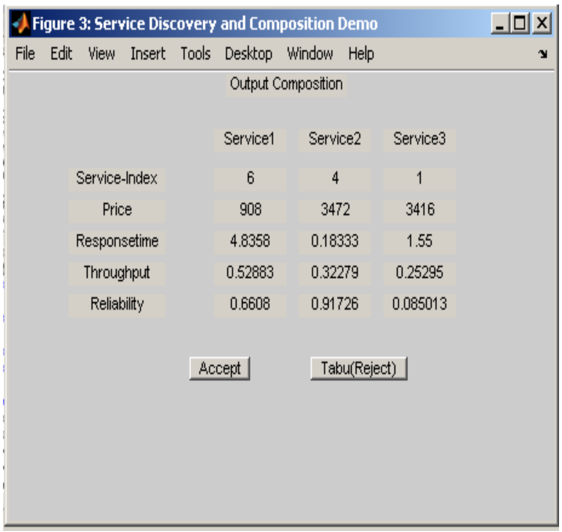
Figure 3: Service Discovery and Composition Demo
File Edit View Insert Tools Desktop Window Help
Accept
Tabu(Reject)

Output Composition

	Service1	Service2	Service3
Service-Index	6	4	1
Price	908	3472	3416
Responsetime	4.8358	0.18333	1.55
Throughput	0.52883	0.32279	0.25295
Reliability	0.6608	0.91726	0.085013

Fig 4. Composition output screen

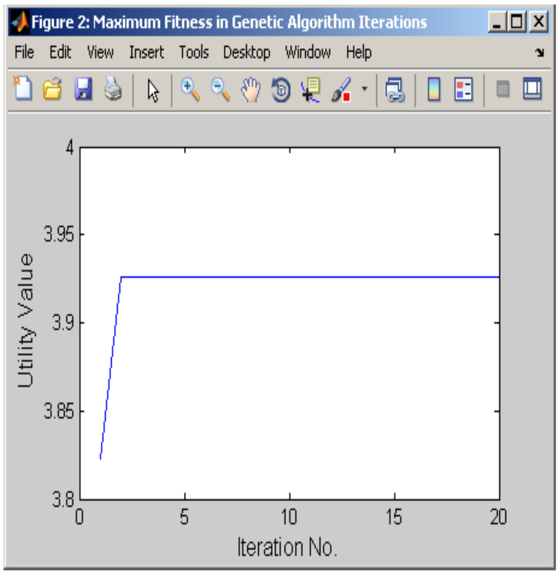

Fig 5. Utility-iteration plot

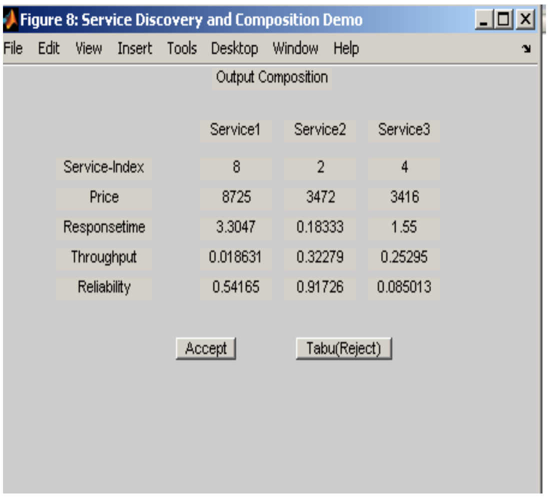
Figure 8: Service Discovery and Composition Demo
File Edit View Insert Tools Desktop Window Help
Accept
Tabu(Reject)

Output Composition

	Service1	Service2	Service3
Service-Index	8	2	4
Price	8725	3472	3416
Responsetime	3.3047	0.18333	1.55
Throughput	0.018631	0.32279	0.25295
Reliability	0.54165	0.91726	0.085013

Fig 6. Service composition after Tabu

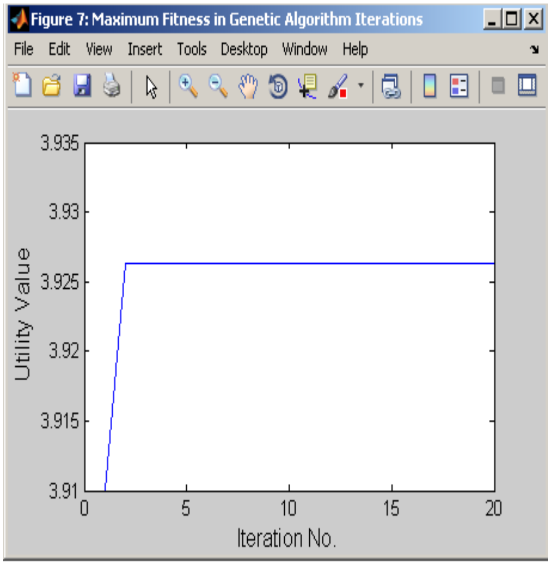

Fig 7. Utility-iteration plot on Tabu

V. Conclusion

A number of approaches are suggested to address the problem of selection of web service composition. Web service composition has become an important issue with an emerging of cloud computing as a new paradigm of providing services to the end user. In this paper our emphasis is on the selection of user-centric cloud service composition based on the weighted priorities required by the end user. we had applied the hybrid GA-Tabu search algorithm for random selection of chromosome/ service providers and to compute the utility of each service composition. The use of Tabu search approach solves the optimization problem and is capable to get escape from trap of local optimality. Tabu search is used to select the candidate service/ chromosome with the best fitness after the crossover and mutation operation by Genetic Algorithm. The approach increases the diversity of population and more search space with scope of improvement in solution. We had evaluated our approach experimentally using randomly generated cloud service QoS datasets. The results proves that the service composition selected using our approach is more user-centric and effective.

References

[1] M. Alrifai, D. Skoutas and T. Risse, "Selecting skyline services for QoS-based web services composition," Proceedings of the 19th International conference on World Wide Web (WWW 2010), 2010, pp. 11-20.

[2] V Cardellini, E. Casalicchio, V.Grassi, and F.Lo Presti, "Flow-Based Service Selection for Web Service Composition Supporting Multiple QoS Classes," Proceedings of the 5th IEEE International Conference on Web Services (ICWS) 2007, 2007, pp.743-750.

[3] A. Goscinski and M. Brock, " Toward dynamic and attribute based publication, discovery and selection for cloud computing," Future Generation Computer Systems, vol. 26, pp. 947-970, 2010.

[4] D. Ardagna and B. Pernici, “Adaptive service composition in flexible processes," IEEE Transactions on Software Engineering, vol. 33 pp.369-384, 2007.

[5] S.Y. Hwang, E.P.Lim, C.H. Lee and C.H. Chen, "Dynamic Web service selection for reliable web service composition," IEEE Transactions on services computing, vol.1, pp. 104-116, 2008.

[6] T. Yu, Y. Zhang, and K.J. Lin, "Efficient algorithm for web service selection with end-to-end QoS constraints," ACM Transactions on the web, Vol. 1, pp. 1-26, 2007.

[7] M. Alrifai and T. Risse, "Combining global optimization with local selection for efficient QoS -aware service composition," Proceedings of the 18th International conference on world wide web (www 2009), 2009, pp.881-890.

[8] P.A. Bonatti and P. Festa, "On Optimal Service Selection," Proc. 14th Int’l Conf. World Wide Web (WWW ’05), pp. 530-538, 2005.

[9] L. Zeng, B. Benatallah, A.H. Ngu, M. Dumas, J. Kalagnanam, and H. Chang, "QoS-Aware Middleware for Web Services Composition," IEEE Trans. Software Eng., vol. 30, no. 5, pp. 311-327, May 2004.

[10] N.N. Liu and Q. Yang, "Eigen rank: A Ranking-Oriented Approach to Collaborative Filtering," Proc. 31st Int’l ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR ’08), pp. 83-90, 2008.

[11] C. Yang, B. Wei, J. Wu, Y. Zhang, and L. Zhang, "Cares: A Ranking-Oriented Cadal Recommender System," Proc. Ninth ACM/IEEE-CS Joint Conf. Digital Libraries (JCDL’09), pp. 203-212, 2009.

[12] L. Wang, J. Shen and J. Yung, "A Survey on Bio-inspired Algorithms for Web Service Composition," Proc. of the 16th International Conference on Computer Supported Cooperative Work in Design, pp.569-574, 2012.

[13] S.Bahdori, S. Kafi, K. Zamani far, M Reza Khayyambashi, "Optimal web service composition using hybrid GA-TABU search," Journal of Theoretical and Applied Information Technology, Vol.9,No.1, pp.10-15,2009.

[14] Jose Antonio Parejo, PabloFernandez, Antonio Ruiz Cortes," QoS-aware services composition using Tabu search and hybrid genetic algorithms," Actas de Tallers de Ingeniera del software y Bases de Datos Vol. 2. No.1, pp.55 - 66, 2008.

[15] S.Wang, Z.Zheng, Q.Sun, H.Zou and F. Yang," Cloud Model for Service Selection," Workshop on Cloud Computing, IEEE INFOCOM 2011, pp.677 - 682, 2011.

[16] J.H.Jang, D.H.Shin and K.H. Lee, "Fast quality driven selection of composite Web Services, "Proceedings of the 4th European Conference on Web Services (ECOWS 2006), 2006, pp.87-96.

[17] Wenying Zeng, Yuelong Zhao and Junwei Zeng, “Cloud service and service selection algorithm research”, GEC Summit, pp 1045-1048. ACM, 2009.

[18] Ahmed F. Ali, "Genetic local search algorithm with self-adaptive population resizing for solving global optimization problems," I.J. Information Engineering and Electronic Business, pp 51-63, 2014.

Authors’ Profiles

Vivek Gaur received a M.E. degree from Birla Institute of Technology and Science (BITS), Pilani, India in December 2001 and pursuing a Ph.D. degree from the Banasthali University, Jaipur, India, since May 2011. Currently, he is an Assistant Professor of Computer Science Department at the BIT, Mesra, Ranchi, India. Mr. .Gaur's major research interest lies in cloud and grid computing, distributed systems, social networking, and web mining technologies.

Praveen Dhyani received a Ph.D. degree from Birla Institute of Technology and Science (BITS), Pilani, India. Currently he is a Professor of Computer Science and Executive Director at Banasthali University Jaipur Campus. Previously Dr. Dhyani established and headed national and international centers of BIT MESRA at Jaipur, Bahrain, Muscat, and RAK (UAE). His R&D accomplishments include electronic devices to aid foot drop patients and development of voice operated wheel chair. He is also a member of the Program me Execution Committee (PEC), UIDAI Biometric Centre of Competence (UBCC), Unique Identification Authority of India (UIDAI), Planning Commission, Government of India.

O.P. Rishi received a M.Tech. degree from Birla Institute of Technology (BIT), Mesra, Ranchi, India in May 1998 and a Ph.D. degree from the Banasthali University, Jaipur, India. Currently, he is an Associate Professor of Computer Science Department at the Kota Engineering College, Kota, India. Dr. Rishi's major research interest lies in service-oriented architecture, distributed systems, web and data mining technologies.

How to cite this paper: Vivek Gaur, Praveen Dhyani, O. P. Rishi,"A GA-Tabu Based User Centric Approach for Discovering Optimal Qos Composition", IJMECS, vol.7, no.2, pp.56-62, 2015.DOI: 10.5815/ijmecs.2015.02.08