Optimal QoS based Web Service Choreography using Ant Colony Optimization

Alexander T
Research Scholar
Dept. of Computer Science
Bharathidasan University
Tiruchirappalli, India.

E. Kirubakaran, PhD
Additional General Manager
SSTP (Systems)
Bharat Heavy Electricals Ltd
Tiruchirappalli, India.

ABSTRACT
Web services have become an integral part of any web based application, due to their availability and ease of use. As the number of web services start increasing uncertainty arises as to which service should be selected. Even though this can be solved by ensuring the appropriate quality of service parameters, performing these checks on numerous services would prove to be a tedious task and time consuming. Hence this paper proposes an efficient QoS based service choreography, that selects the web services on the basis of the quality parameters and cost. A modified Ant Colony Optimization is used for this purpose. The modification is brought about by modifying the evaporation rate of each of the links depending on certain parameters. An effective result that satisfies the QoS constraints is obtained within the stipulated time.

Keywords: Web Service Choreography; QoS based service selection; ACO

1. INTRODUCTION
A Web Service is a method of communication between two devices in a network.

The W3C defines a Web service as:

"*a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards.[5]*"

Due to the explosion of internet and the increase in the usage of internet by the masses necessity of web based interactions have increased, which automatically led to the increase in the amount of web services. Performing a task online has never been cheaper and more efficient with the gem of the crown being explosive number of choices to choose from. This positive aspect also has its own downsides. The availability of many services for a single task leads to an uncertainty during the service selection process.

Orchestration is an integration mechanism which selects the services necessary for building the workflow. The services need not necessarily be sequential. It is very powerful and is strongly supported by many vendors. Orchestration usually refers to an executable inter-organizational process that is provided by one party. This executable process interacts both with external services of the other partners and with internal services. Language used for definition of orchestrations is WSBPEL.

Choreography is an integration mechanism that describes message sequences between different parties, and conditions and constraints under which messages are exchanged in an inter-organizational business process. Choreography can be described from a global, and a local perspective. The global model of choreography specifies the message exchanges from an overall point of view. The local choreography model defines the message interactions from the perspective of one party. While choreographies serve as interaction contract between business partners, they are not meant to be executed by themselves. Choreography model is described using XML-based WS-CDL language. Since it is a global model of collaborations between all involved parties, each involved party can use it to build and test solutions that conform to this global model. Choreography can be seen as a coordination mechanism between atomic Web services.

The evolutionary algorithms are a class population based meta-heuristic techniques that are inspired by the biological evolution of the organisms. Here, the individuals are represented by the candidate solutions, while their quality is determined using the fitness function. The fitness function determines which organism passes on to the next generation. The individuals are modified by repeated application of the operators. The flexible nature of the evolutionary algorithms to adapt to the changing environment comes as an added advantage. Further, an optimal or near optimal routing solution would suffice. Due to the basic nature of the problem (path finding), any heuristic or meta heuristic method that performs graph processing will prove to be a good choice. Hence for our application, we have chosen Ant Colony Optimization (ACO).

The Ant Colony Optimization (ACO) was proposed by Marco Dorigo [6]. This method was based on the behavior of ants during their search for food. Every ant moves through a path and finally finds the food source. On its way back, it leaves a pheromone trail marking the path it has travelled as one of the available paths for reaching the food source. This can be in simpler terms a graph traversal algorithm, finding optimal routes from a source to the destination. Trail intensity in a path (amount of pheromone deposited) and Visibility of a path (inversely proportional to the distance) plays a vital role in the selection of a path. In due course, the best trail that represents the shortest path prevails and the other paths tend to fade away due to the property of evaporation. Several variants of ACO have been proposed [7], [8], [9], [10] and [11]. These variants differ in terms of the amount of pheromone deposit and the phase during which the pheromone deposition mechanism takes place (either after obtaining a complete path, or after the selection of every node).

The remainder of this paper is structured as follows; Section 2 provides the related works, Section 3 presents overall system architecture, Section 4 presents a detailed discussion on the working of the QoS based Web Service Choreography technique, Section 5 presents the results and Section 6 concludes the study.

2. RELATED WORKS

A multi constrained QoS aware service selection algorithm for Wireless Mesh Networks is presented in [12]. The ant movement begins from the source node, unlike the usual ACO in which ants can be launched from any node in the graph. Next node selection is based upon the cost metric and the constraints are chosen both from the user's perspective and from the network's perspective. It uses the technique of Guided Search Evaluation (GSE). It works in three layers, the service request and generation layer, service request and constraint satisfaction layer and the service processing layer. This mechanism fails to address the issue about other QoS constraints except for the cost, hence might not provide an efficient solution.

An analysis of Stochastic ACO (S-ACO) is presented in [13]. A pre-test is carried out on S-ACO using the probabilistic Travelling Salesman Problem. More comprehensive tests were conducted using TSP with time windows with stochastic service times. Stochastic Simulated Annealing was used as a comparison to determine the accuracy of the results. It was found that S-ACO outperformed Stochastic Simulated Annealing in most of the scenarios. This paper also discusses the fine-tuning of S-ACO depending on the problem variety.

A modified biologically inspired variant of ACO was presented in [14]. In ACO, the ants usually navigate based on the pheromone trails on the path. In [14], a simple modification is brought about by providing the ants with an additional sense of smell. Addition of this behavioral capacity makes ants more intelligent. The behavior to sense smell is not just limited to the food from the source. Hence this also helps in the determination of efficient and shortest paths in the provided graph.

A heuristic based Ant Colony Optimization Algorithm is described in [4]. It uses the principle of Marco Dorigo, proposed in 1992 [6]. It modifies the algorithm described in [6] to incorporate multiple properties as conditional considerations to find the optimal route for the travelling salesman problem. Every property is related to the evaporation rate with proportionality constant. According to the behavior, every property is either directly or inversely proportional to the evaporation rate, and in turn with the pheromone trail, and in turn with the probability of being selected.

A resource constrained scheduling problem with strict deadlines is described in [15], which can be solved by a hybrid approach involving ACO. Two deadlines, soft and hard; also called the preferred and final deadlines are used here as the basic constraints. The basic necessity is to reduce the cumulative tardiness. This resource constrained problem uses a hybrid approach involving ACO, beam search and constraint programming.

An ACO based service composition method is presented in [16]. Service composition is usually performed by integrating the required services by selecting them depending on certain QoS or transactional constraints. The users can specify these constraints and preferences. Depending on the constraints, a directed acyclic graph is constructed with the shortlisted services. ACO is applied on this graph to provide the necessary results. [17] describes the importance of service composition, and the role it plays in providing interoperability.

3. SYSTEM ARCHITECTURE

A task in a web application can be performed by using the available web services. But, the downside here is that many web services are available that can complete the given task, but with different QoS constraints and cost. The challenge her is to select a web service that performs the required operation meeting the standards of the current QoS constraints, and within the given cost. Jobs in a web application can usually be divided into many such tasks, hence the real challenge is to perform efficient service choreography, with each selected service meeting the QoS requirements and under the specified cost. Figure 1 shows the system architecture of the QoS based Service Choreography.

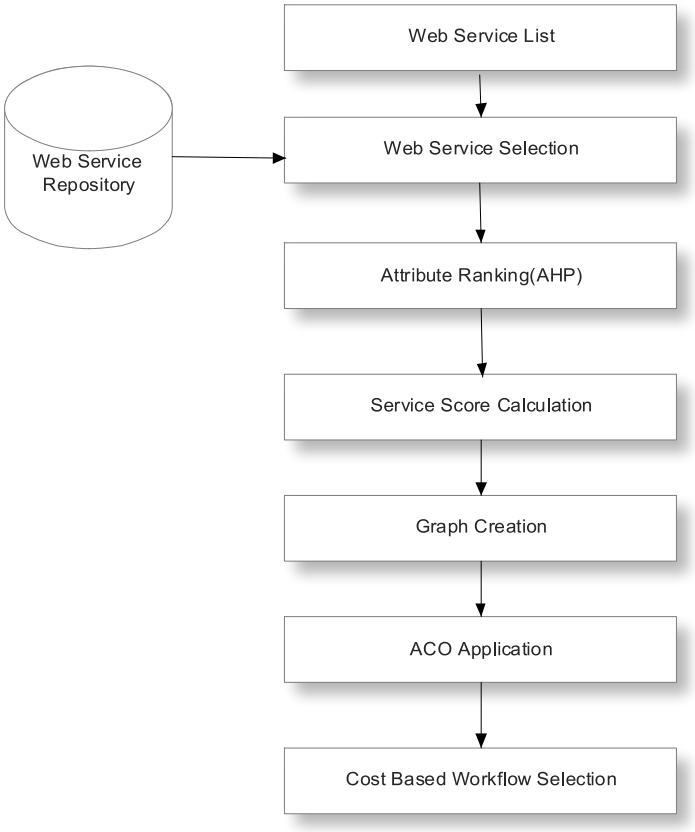

Fig 1: Web Service Choreography based on ACO

The process begins with analyzing various tasks to be performed for completing the job. All web services pertaining to the task in hand are selected from the web service repository. QoS parameters of the selected web services are taken and parameter ranking is performed using AHP. The rank values are used to calculate the overall score of each web service. A directed graph is then created, on which ACO is obtained to determine the appropriate web services that cater to the needs of the current application.

4. OPTIMAL QOS BASED WEB SERVICE CHOREOGRAPHY USING ANT COLONY OPTIMIZATION

4.1. Web Service Selection Phase

The principal work of this phase is to analyze the job description, divide them into tasks and to obtain the web services that correspond to these tasks. The input is passed on to the Neural Network based keyword analyzer. The input provided is in the form of distinct entities. The input layer of the Neural Networks reads the entries and passes them to the processing phase. Data analysis is performed in the processing phase. Keyword division and elimination of commonly occurring words is performed. Redundancies are checked, and eliminated. The remaining words are used to determine the corresponding web services [1].

$$S = \begin{bmatrix} w_1/w_1 & w_1/w_2 & \cdots & w_1/w_n \\ w_2/w_1 & w_2/w_2 & \cdots & w_2/w_n \\ \vdots & \vdots & \ddots & \vdots \\ w_n/w_1 & w_n/w_2 & \cdots & w_n/w_n \end{bmatrix} = \begin{bmatrix} 1 & s_{12} & \cdots & s_{1n} \\ s_{21} & 1 & \cdots & s_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ S_{n1} & s_{n2} & \cdots & 1 \end{bmatrix} \quad (1)$$

Direct weight values (9 point scale) can also be assigned to the attributes. But this method is to be adopted only if the user is confident of the weight being assigned.

4.2. Web Service Processing Phase

Every task in the job corresponds to multiple web services, hence we use QoS constrains to determine the appropriate web service. Attribute ranking is performed using AHP and the overall score for each web service is obtained.

Service rankings are performed using the weighted sum model of relevant measures. Default weights are set for each attribute. These weights can be altered by using the pair-wise comparison method proposed by Saaty [2,3] or direct arbitrary weight. Analytic Hierarchy Processing is used for performing the comparison process [1].

The pairwise comparisons are made depending on a 9 point scale. In the pairwise comparison matrix, each value represents the relative importance of the component on row (u) over the component on column (v). The reciprocal value of the expression is used when the component v is more important than the component u. The comparison matrix S is defined as

The overall score (rank) of the shortlisted web services is obtained by using the formula,

$$w_{ir} = \sum w_{ia} * w_a \quad (2)$$

Where w_{ir} is the rank of the web service *i*, w_{ia} is the normalized attribute value of the web service *i* and w_a is the weight of the attribute. Since the attributes in a web service contains a values under different ranges, they are first normalized and then the service rank is calculated.

4.3. Graph Building Phase

A directed graph is created in this phase, which is used by ACO to determine the web services that are to be used in the final workflow. Every web service that has been selected is considered to be a node. The graph is divided into phases depending upon the number of tasks that are to be performed. Every phase is constructed with web services that perform the particular task. Every node in a phase is connected by a directed edge to all the services that are present in the next phase. The weight of an edge is the rank from which the edge originates.

Consider that an application contains 5 tasks. There are 4,3,3,1 and 2 web services available to perform tasks 1,2,3,4 and 5 respectively. A graph for such a scenario looks like (Figure 2),

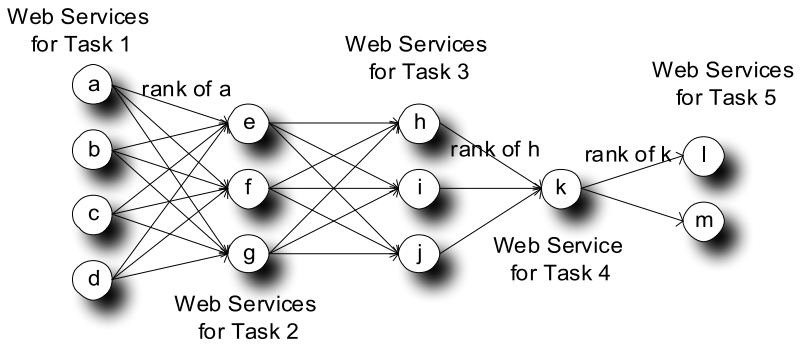

Fig 2: A sample web service scenario graph (with sample edge weights)

4.4. ACO Application

ACO is a graph traversal algorithm that tries to find the shortest path covering all the nodes in a graph. Since it is a metaheuristic based method, it is not guaranteed that we obtain the shortest path every time the algorithm is executed. Instead, the promise of the algorithm is that it can return an optimal path during every execution within a fixed time. Even though the available statistical methods return the best results, they do it at the expense of time. The time taken by these algorithms cannot be determined and mostly it is above the bearable threshold. Time is a very important factor in every application, the expectation of every application is to perform tasks within a threshold time, and the readiness of the application to accept a near optimal solution has led to the development of the metaheuristics. Since our application comes under this category, it would be efficient to use ACO rather than a statistical method.

Even though the basic working of an ACO is to traverse all the available nodes in a graph, our application requires traversal of one node from each phase and the sequence of travel should be maintained. Hence the graph used for our application is directed, unlike the usual undirected graph used in ACO.

A predefined set of ants are released into the graph at the starting point. Every ant is free to choose the node it starts from. The initial node determination is performed using a Normal Distribution function. A discussion on the selection of the number of ants and how this determines the quality of the final solution is discussed in Section 5. In our sample scenario (Figure 2), four choices (a,b,c,d) are available from which an ant can begin its execution. Every ant then determines the next node of traversal using the probability factor.

$$p_{ij}(t) = \frac{[\tau_{ij}(t)]^{\alpha}.[\eta_{ij}]^{\beta}}{\sum_{j=1}^{n}[\tau_{ij}(t)]^{\alpha}.[\eta_{ij}]^{\beta}} \qquad (3)$$

Where $P_{ij}(t)$ is the probability of traversing from node *i* to node *j*, the trail intensity (τ_{ij}) in the route between points i and j, η is the visibility (since in this application, we consider deriving the route with a higher QoS, the visibility is w_{ir} and not $1/w_{ir}$ as in the conventional ACO), the values of amount of pheromone to be deposited (Q), importance of trail intensity (α), importance of visibility (β), evaporation parameter (ρ) and time (t).

After the selection of the destination point (j), a pheromone trail of quantity Q is left behind is defined by the formula,

$$\tau_{ij}(t+1) = \rho.\tau_{ij}(t) + \Delta\tau_{ij}(t, t+1) \qquad (4)$$

During the initial run, a constant trail is places on all the nodes of the graph. This trail is then modified by the evaporation factor (ρ) and the trail update after each run. The current proposal uses the Ant Density model which does not consider the distance as one of its factors while depositing pheromones. The value of $\Delta\tau_{ij}$ is calculated as,

$$\Delta\tau_{ij}^{k}(t, t+1) = \begin{cases} Q & if\ ant\ k\ goes\ from\ i\ to\ j\ between\ t\ and\ t+1 \\ 0 & otherwise \end{cases} \qquad (5)$$

The evaporation rate (ρ) is determined by the web service attributes. The service attributes used in our dataset are Response Time (λ_{rt}), Availability (λ_a), Throughput (λ_t), Successability (λ_s), Reliability (λ_r), Compliance (λ_c), Best Practices (λ_{bp}) and Latency (λ_l).

Statement: *Given a graph G(V,E), where V is the set of vertices that corresponds to the web services and E is the edges or connections defining the sequence between two nodes.* τ_{ij} *represents the pheromone concentration,* ρ_{ij}

represents the evaporation parameter, λ_{rt} represents the response time, λ_a represents the availability, λ_t represents the throughput, λ_s represents the successability, λ_r represents the reliability, λ_c represents the compliance, λ_{bp} represents the best practices and λ_l represents the latency then the probability of selection of the edge ij by an ant is directly proportional to λ_{rt} and λ_l, and is inversely proportional to λ_t, λ_a, λ_s, λ_r, λ_c and λ_{bp}.

Proof:

According to equation (3),

$$P_{ij} \alpha \tau_{ij} \quad (6)$$

The probability of an edge to be selected is directly proportional to the pheromone deposits τ_{ij} in that edge.

The pheromone deposit in a particular edge is determined by two factors; the number of ants that had traversed that path, and the evaporation rate ρ_{ij}.

$$\tau_{ij} \alpha \frac{1}{\rho_{ij}} \quad (7)$$

By default the evaporation rate is defined initially and does not change, while in our paper, every edge has different evaporation rates depending on their attributes. This helps in faster and more convergence. The evaporation rate of an edge is determined by the following equations

$$\rho_{ij} \alpha \lambda_{rt} \quad (8)$$

$$\rho_{ij} \alpha 1/\lambda_a \quad (9)$$

$$\rho_{ij} \alpha 1/\lambda_t \quad (10)$$

$$\rho_{ij} \alpha 1/\lambda_s \quad (11)$$

$$\rho_{ij} \alpha 1/\lambda_r \quad (12)$$

$$\rho_{ij} \alpha 1/\lambda_c \quad (13)$$

$$\rho_{ij} \alpha 1/\lambda_{bp} \quad (14)$$

$$\rho_{ij} \alpha \lambda_l \quad (15)$$

Equations (8) to (15) shows the relationship between the pheromone deposit and the attributes. The evaporation rate increases as $\lambda_{rt} and\ \lambda_l$ increase, which implies that these properties are directly proportional to the evaporation rate. Hence by equation (7),

$$\rho_{ij} \alpha \lambda_{rt}.\lambda_l \quad (16)$$

The evaporation rate decreases as the values $\lambda_a, \lambda_t, \lambda_s, \lambda_r, \lambda_c\ and\ \lambda_{bp}$ increase, which implies that these properties are inversely proportional to the evaporation rate. Hence by equation (7),

$$\rho_{ij} \alpha \frac{1}{\lambda_a.\lambda_t.\lambda_s.\lambda_r.\lambda_c.\lambda_{bp}} \quad (17)$$

By using (7) and combining equations (16) and (17),

$$\tau_{ij} \alpha \frac{\lambda_a.\lambda_t.\lambda_s.\lambda_r.\lambda_c.\lambda_{bp}}{\lambda_{rt}.\lambda_l} \quad (18)$$

From equation (6), (18) can be written as

$$p_{ij} \alpha \frac{\lambda_a.\lambda_t.\lambda_s.\lambda_r.\lambda_c.\lambda_{bp}}{\lambda_{rt}.\lambda_l} \quad (19)$$

From equation (19) we can imply the relationship between various properties used in the system with respect to P_{ij} (Probabilty of selecting a route from i to j).

Every node *j* connected to *i* will have an associated probability. The probability values determine the node that is to be taken as the successor. After selection, the node is added to the tabu list. The tabu list stores the list of visited nodes for each ant. The size of a tabu list in general is the number of cities in the graph, but in this case, it is the number of phases contained in the graph, or the number of web services required in the workflow. The workflow is determined when the tabu list is full. The best route represents a web service sequence that provides the best quality parameters at the required cost.

5. RESULTS AND DISCUSSION

A simulation is conducted with the QWS dataset (http://www.uoguelph.ca/~qmahmoud/qws/). This dataset consists of 365 Web services each with a set of nine Quality of Web Service (QWS) attributes that have been measured using commercial benchmark tools. Each service was tested over a ten-minute period for three consecutive days. It contains 2507 web service invocations with redundant entries invoking service of same name with different QoS parameters.

The simulations were carried out using 16 different sets of alpha and beta values (Table 1), and varying the rho (0.2,0.5,0.7) and the number of ants. Five sets of ants were chosen, where 2 sets contain values below the total number of phases (number of required services for the workflow), a set equal to the number of phases and two sets greater than the number of phases.

Table 1: Parameter Sets

Set No	Alpha	Beta
Set 1	**0.5**	**1**
Set 2	**0.5**	**2**
Set 3	**0.5**	**5**
Set 4	**0.5**	**10**
Set 5	**1**	**1**
Set 6	**1**	**2**
Set 7	**1**	**5**
Set 8	**1**	**10**
Set 9	**1.5**	**1**
Set 10	**1.5**	**2**
Set 11	**1.5**	**5**
Set 12	**1.5**	**10**
Set 13	**2**	**1**
Set 14	**2**	**2**
Set 15	**2**	**5**
Set 16	**2**	**10**

A consolidation of all the graphs was used for analysis. Figure 3 shows a comparison between the number of ants and time taken for finding the final solution. It can be observed that the graph increases linearly, which is more likely as the number of ants tended to increase the number of processes.

Figure 4 shows a comparison between the number of ants and the delivered QoS. As far as a web service is concerned, the higher the obtained QoS, the better. Due to the presence of many graph plots, it becomes difficult to provide the analysis. Hence some of the lines have been removed from the graph (Figure 5).

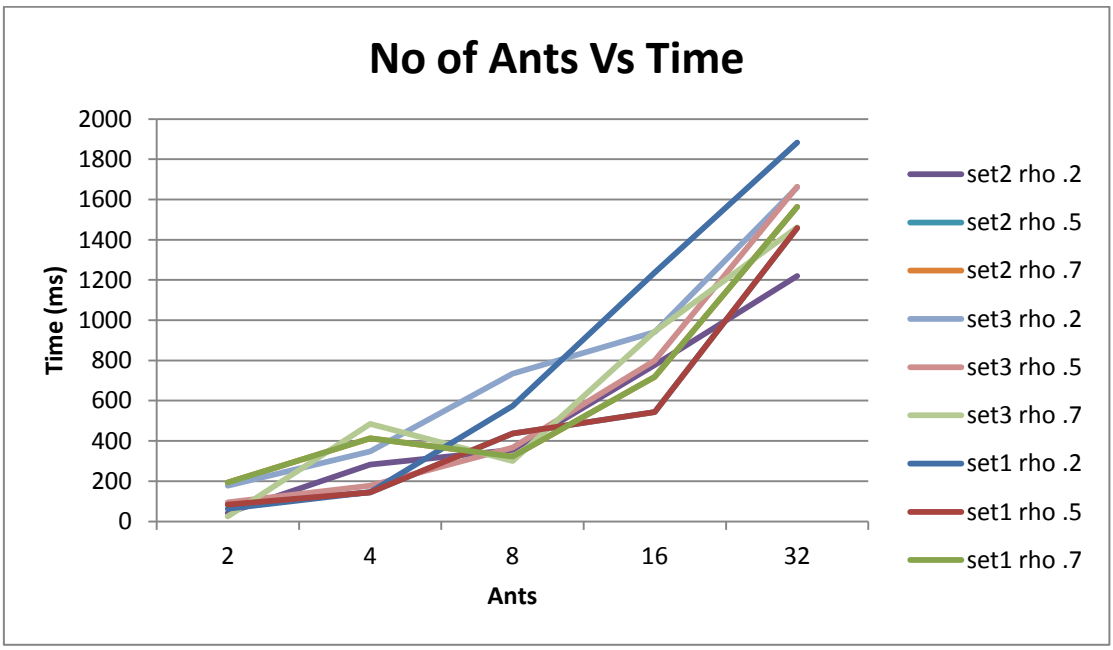

Fig 3: No of Ants Vs Time

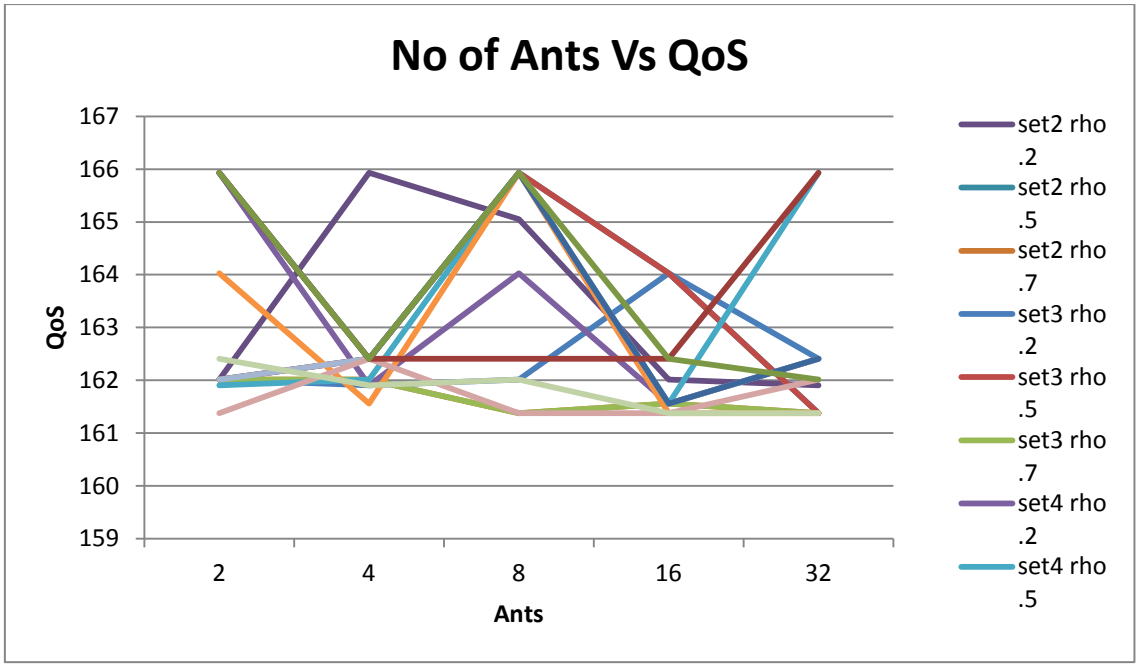

Fig 4: No of Ants Vs QoS

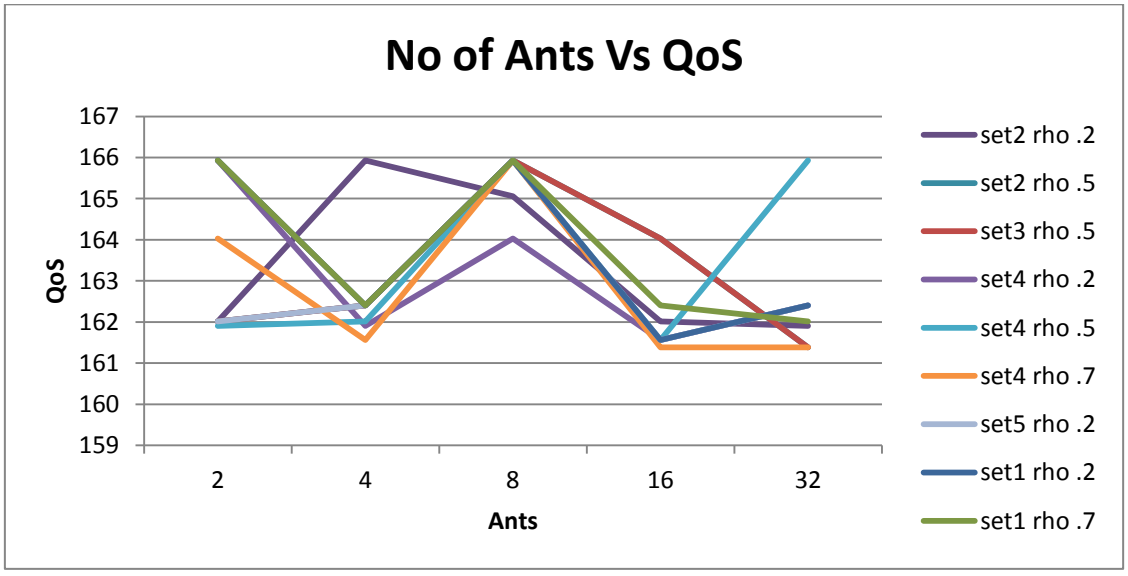

Fig 5: No of Ants Vs QoS (A Closer Analysis)

From the legend entries, it can be clearly seen that the contribution of a rho value of 0.7 is very least followed by 0.5. The beast entries were provided by entries with a rho value of 0.2 (A very few sets of parameters are shown in the figures in order to provide clarity in the graphs. Actual analysis was carried out using all the 12 sets of parameters). The contribution of alpha and beta does not vary the results to a considerable extent.

Further analysis reveals that the best results are obtained within the boundaries of the total number of phases. The simulation was carried out using 7 phases. It can be clearly seen in Figure 5 that most of the maximum convergence occurs under the boundaries of 6-9 ants.

6. CONCLUSION

The results show that the algorithm implementing ACO has better convergence time and provides efficient QoS to the user. It can be clearly seen that from the point of view of the quality of service, our method is in par with other approaches, and when it comes to the time required for estimating the service workflow, average time of convergence is less than a second. This can be further improvised by integrating it with the cost constraints. Further enhancements can be provided by adding dynamic attributes and requirements to the system. The ACO algorithm, being inherently stochastic has a high probability of working efficiently in such scenarios.

7. REFERENCES

[1] Alexander T, Kirubakaran E. 2013. Neural network and GA based intelligent b2b negotiation system. International Journal of Computer Applications. Volume 68 - Number 17. Doi: 10.5120/11668-7264.

[2] Saaty T.L. 1980. The analytic hierarchy process: planning, priority setting, resources allocation. Publisher: McGraw-Hill.

[3] Saaty T.L. 2005. Theory and applications of the analytic network process: decision making with bene_ts, opportunities, costs, and risks. RWS publications.

[4] Talreja S. 2013. A Heuristic Proposal in the Dimension of Ant Colony Optimization. Department of Applied Mathematics. Applied Mathematical Sciences, Vol. 7, 2013, no. 41, 2017 - 2026.

[5] Web Services Glossary. W3C. February 11, 2004. Retrieved 2011-04-22.

[6] Dorigo M, Maniezzo V, Colorni A. 1996. Ant system: optimization by a colony of cooperating agents. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on 26.1: 29-41.

[7] Sorin C. Negulescu, Constantin Oprean,Claudiu V. Kifor, Ilie Carabulea. 2008. Elitist ant system for route allocation problem.World Scientific and Engineering Academy and Society (WSEAS) Stevens Point, Wisconsin, USA.

[8] T. Stützle et H.H. Hoos. 2000. Max Min Ant System. Future Generation Computer Systems, volume 16, pages 889-914.

[9] Bernd Bullnheimer Richard F. Hartl, Christine Strau. 1997. A New Rank Based Version of the Ant System - A Computational Study.Working Paper No. 1.

[10] Xiao-Min Hu, Jun Zhang, Yun Li. 2008. Orthogonal Methods Based Ant Colony Search for Solving Continuous Optimization Problems.Journal of Computer Science and Technology. Volume 23, Issue 1, pp 2-18.

[11] Gupta.D.K, Arora, Singh.U.K, Gupta.J.P. 2012. Recursive Ant Colony Optimization for estimation of parameters of a function. Recent Advances in Information Technology (RAIT), International Conference, March 2012, DOI: 10.1109/RAIT.2012.6194620, pages 448 – 454.

[12] Kumar, Neeraj, et al. 2011. An ant based multi constraints QoS aware service selection algorithm in wireless mesh networks. Simulation modelling practice and theory 19.9: 1933-1945.

[13] Gutjahr, Walter J. 2004. S-ACO: An ant-based approach to combinatorial optimization under uncertainty. Ant colony optimization and swarm intelligence. Springer Berlin Heidelberg. 238-249.

[14] Rivero, Jessica, et al. 2012. Using the ACO algorithm for path searches in social networks. Applied Intelligence 36.4: 899-917.

[15] Thiruvady, Dhananjay, et al. 2013. Constraint-based ACO for a shared resource constrained scheduling problem. International Journal of Production Economics 141.1: 230-242.

[16] Wu, Quanwang, Qingsheng Zhu. 2013. Transactional and QoS-aware dynamic service composition based on ant colony optimization. Future Generation Computer Systems 29.5: 1112-1119.

[17] Rajeswari M. et al. 2014. Appraisal and analysis on various web service composition approaches based on QoS factors. Journal of King Saud University-Computer and Information Sciences 26.1: 143-152.