Hybrid ANN approach for Web Service Selection

R.Veeranjaneyulu[1],

Research Scholor,

Department of Computer Science & Engineering,

Sathyabama University, Chennai, India

Dr.P.Venkatraman[2]

Professor,

Department of Computer Science & Engineering,

Sathyabama University, Chennai, India.

ABSTRACT

Earlier, web services are used to retrieve data for small amount of user only, but now a day's more data retrieved through web services by more number of users in all the actions of day to day life. There are numerous web services available in the internet to collect the data. Today internet plays a vital role in every body's life. Internet provides all the information one can access through the web service. It will be processed in such a way that optimal result is obtained. If many users refer the optimal site then server loading will occur to reduce that problem server redirection is used in this project

In this Paper, Neuro-GA approach has been applied for web service selection. The appropriate weight for ANN is generated with the help of genetic algorithm. Neural Networks can recognize only those pattern that network have already learned. It can not recognize the new patterns. The network needs to be trained sufficiently to extract and determine general features which applies to both training and testing data. However overtraining of neural networks lead to the undesired result. BPN uses gradient descent method to minimize the error and update the weights, so there is a probability to stuck in a local minimum. Genetic algorithm (GA) is an optimization technique applied to find exact or approximate solution. The solution generated by GA may or may not be the best solution. Genetic algorithm intensively used to search the solution space. GA requires population size, selection rate, initial weight range, number of training epochs to find the solution of problem. In neuro GA approach, GA is used to generate weight, which is fed to the neural network. GA is continued to find the weight set until the fitness values for 95% of the chromosomes are same. Chromosome is the collection of genes (weights).

Keywords— *Web services, Quality of Service, Service Selection, Genetic Algorithm, Memetic Algorithm, Swarm Intelligence*

1. INTRODUCTION

WEB SERVICE SELECTION ALGORITHMS:

The problem of service selection on the web consists of having an efficient algorithm that can match multiple service consumers and service providers efficiently, while optimizing multiple objectives (QoS parameters). Multiple clients requesting similar services should be satisfied, and secondly, the assignment process of the service consumers and the service providers should be optimized. Please note that one consumer can only be matched with one provider. Since we have several service consumers and equally numbered service providers, the aim is to match the consumer–provider pairs as closely as possible using a Genetic Algorithm (GA), Memetic Algorithm (MA) and Particle Swarm Optimization (PSO) approach.

The QoS criteria in the context of services are execution price, execution time, reliability, reputation, and availability. The values of these QoS parameters range between 0 to 1. Each consumer provides the QoS values based on its requirement of how the request must be executed, and each service provides the value based on its task execution capability. The service provider has a value for each QoS parameter. The service consumer requests a service provider specifying an upper and lower value for each QoS parameter, whereby for some QoS attributes the lower or upper bound is preferred. In particular, the lower bound is preferred for execution price and execution time, and the upper bound is preferred for reliability, reputation and availability.

A. Genetic Algorithm GA is a global optimization algorithm that models natural evolution. In GA, individuals form a generation. An individual corresponds to one match. The match is implemented as a vector, which is also referred to as a chromosome. Dimensions in the vector correspond to providers, and values correspond to consumers. Therefore, if the vector has value 2 at its 4th position (dimension), consumer 2 is matched with provider 4. Every number representing a consumer can only be at one position in the vector; otherwise, the vector represents a non-valid match. At the beginning, the first population is randomly initialized. After that, the fitness of the individuals is evaluated using the fitness function. After the fitness is evaluated, individuals have to be selected for paring. The selection method used is tournament selection. Always two individuals are paired, resulting in an offspring of two new individuals. In the pairing phase, a random crossover mask is used, i.e. the positions (dimensions) for which crossover occur are selected randomly. If crossover occurs at certain positions (dimensions), individuals that are mated exchange their values at that position and the resulting individuals are used as offspring. The crossover has to make sure that the offspring presents a valid match. Therefore, if two values are exchanged, other positions in the two match vectors are usually effected as well. The offspring faces mutation with a certain low probability. After mutation, the fitness of the offspring is calculated. Then, either all individuals from the last generation compete against the whole offspring, or the offspring only compete with its corresponding parents. In this implementation, all individuals from the old generation compete with all individuals in the new generation. In order to implement this, all individuals are ordered by their fitness score, using a non-recursive advanced quick sort algorithm, which after sorting truncates the lower half. After the new generation is selected, the GA will start over, and continue with parent selection and crossover until a certain number of iterations are reached

B. Memetic Algorithm Evolutionary algorithms are not well suited for fine-tuning the search, in particular in complex combinatorial spaces, and therefore, researchers have developed hybridization methods to overcome this problem and to improve the efficiency of the search [5]. The combination of using evolutionary algorithms as well as local search techniques was named Memetic Algorithms (MAs). MAs are basically an extension of evolutionary algorithms that apply a separate process to refine solutions by improving the fitness of the individuals with methods such as hill-climbing or simulated annealing. MAs were inspired by the models of adaptation in natural systems, in particular the combination of the evolutionary adaptation of a population with individual learning. GAs on the one hand and local search on the other hand, are captured within MAs, thus rendering a methodology that balances well between generality and problem specificity. The name of MAs was inspired by Richard Dawkins‟ concept of a meme, which represents a unit of cultural evolution that can exhibit local refinement [6]. A meme represents a learning or development strategy [7]. Memetic algorithms are also known as Hybrid Evolutionary Algorithms [8], Baldwin an Evolutionary Algorithms [9], Lamarckian Evolutionary Algorithms [10], Cultural Algorithms or Genetic Local Search. All techniques combine local search heuristics with the evolutionary algorithms‟ operators. Combinations with constructive heuristics or exact methods may also be considered within this class of algorithms. In this research, we apply an exact method for the local search. For some problem domains, MAs have been shown to be both more efficient and more effective than traditional evolutionary algorithms with regards to requiring orders of magnitude fewer evaluations to find optima, and identifying higher quality solutions. In particular, for many combinatorial optimization problems, where large instances have been solved to optimality, and where other meta-heuristics have failed to produce comparable results, such as the quadratic assignment problem and the traveling salesman problem, MAs have proven themselves to be very effective.

2. SWARM INTELLIGENCE

Swarm intelligence is the collective behavior of a group (swarm) of animals as single living creatures where collective intelligence emerges via grouping and communication. When the route of a swarm of ants is blocked, it can be observed that they find another new shortest route to their destination; this is robustness. These agents (ants) can be added or removed without compromising the total system due to its distributed nature, this is reliable. Single parts may be break down without impairing the overall system, these complex systems are convenient to work because of simplicity of their individual parts. There are two popular swarm inspired methods: Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO).

A. Particle swarm optimization: Overview Particle Swarm Optimization (PSO) is a population based stochastic optimization technique developed by Eberhart and Kennedy in 1995 inspired by the social behavior of flocks of birds and school of fish. In particle swarm intelligent system, bird flocking behavior is simulated for optimization problems. The concept is simple, has few parameters, is easy to implement, and has found applications in many areas. Consider the scenario, group of birds are randomly searching for food in the area. Only one particular location of the area being searched contains food. None of the birds knows where the food is. The sole knowledge they have is how far the food is in each iteration. So, the best and most effective strategy to find food is to follow the bird that is nearest to it. Each single solution is a „bird" in the search space and can be treated as a „particle". Each particle has fitness value. The fitness function to be optimized is evaluated for every particle, and has velocities which direct the flying of the particles. By following the currently optimum particles, the particles fly through the problem space. In PSO, each particle is initialized with a group of random particles, which are solutions; optima are searched for by updating subsequent generations. In every iteration, each particle is updated by following two best values. The first best value is the best solution or fitness it has achieved so far. The variable pbest contains the best fitness value. The best value obtained so far by any particle in the population which is tracked by the particle swarm optimizer is global best and called as gbest. These best solutions are obtained from a relation maintained by the current particle"s velocity and position [13].

B. Particle swarm optimization: Algorithm and Pseudo Code

1) Algorithm

v[] = v[] + c1 * rand() * (pbest[] - present[]) + c2 * rand() * (gbest[] - present[])

present[] = persent[] + v[]

Where v[] is the particle velocity, persent[] is the current particle (solution). pbest[] and gbest[] are defined as stated before.
rand () is a random number between (0,1). c1, c2 are learning factors. usually c1 = c2 = 2.

2) Pseudo Code

```
For each particle
    Initialize particle
END

Do
    For each particle
    Calculate fitness value
    If the fitness value is better than the best fitness value (pBest) in history
    set current value as the new pBest
End
Choose the particle with the best fitness value of all the particles as the gBest
    For each particle
    Calculate particle velocity according equation
    Update particle position according equation
End
```

3. PROPOSED WORK FOR WEB SERVICE SELECTION:

The following sub-sections highlight on the classification methods used for Web Service selection.

3.1 Genetic Algorithm:

In this approach, genetic algorithm is used for weight determination for neural network. Neural network having `x' number of input nodes, `y' number of hidden nodes, and `z' number of output nodes. Output of the input layer is fed to the input. It is given by the lowing equation.

$$O_{in} = I_{in} \quad \textbf{3.1}$$

Sigmoidal function is used as activation function at hidden and output layer. The output of hidden layer **`O'** $_h$ for the net input **`I'** $_{inh}$ can be given as:

$$O_h = \frac{1}{1+e^{-I_{inh}}} \quad \textbf{3.2}$$

The output of output layer **`O'**$_0$ o for the net input of the output layer **`O'** $_{ink}$ is represented as:

$$O_o = \frac{1}{1+e^{-o_{ink}}} \quad \textbf{3.3}$$

The number of weight that is to be determined for neural network can be calculated as**:**

$$N = (x+z)*y \quad \textbf{3.4}$$

Each weight is equivalent to gene which is randomly generated. each gene being a real number. Each gene is coded as binary digit and let the number of digits in weights is l. The length of the chromosome S can be computed as:

$$S = N*l = (x+y)*y*l \quad \textbf{3.5}$$

Fitness value for each chromosome is determined after weights are extracted from each chromosome using the following equation: The weight equivalent to each gene can be evaluated using following equation.

$$W_j = \begin{cases} -\frac{t_{jl+2}*10^{l-2}+t_{jl+3}*10^{l-3}+....+t_{(l+1)l}}{10^{l-2}} & \text{if } 0 <= t_{jl+1} < 5 \\ +\frac{t_{jl+2}*10^{j-2}+t_{jl+3}*10^{l-3}+....+t_{(l+1)l}}{10^{l-2}} & \text{if } 5 <= t_{jl+1} <= 9 \end{cases} \quad \textbf{3.6}$$

The fitness value for each chromosome is derived with the help of fitgen algorithm which can be given as.

3.2 Algorithm to compute fitness function: FITGEN()

Input: $I_i = (I_{1i}, I_{2i}, I_{3i.....} I_{li})$

Output: $T_i = (T_{1i}; T_{2i}; T_{3i} \ldots\ldots T_{ni})$

where I_i, T_i represent the input-output pairs of network.

Step 1: Weights Wj from Ch_i are calculated using equation

Step 2: Considering Wj as a constant weight, the network is trained for N input instances and the estimate value O_{in} is found.

Step 3: For each input instance k error E_K can be evaluated using the following equation:

$$|E_k = (T_{ki} - O_{ki})^2$$

Step 4: Root mean square error (RMSE) for the chromosome Ch_i is given by the following equation:

$$E_i = \sqrt{\frac{\sum_{k=1}^{k=N} E_k}{N}}$$

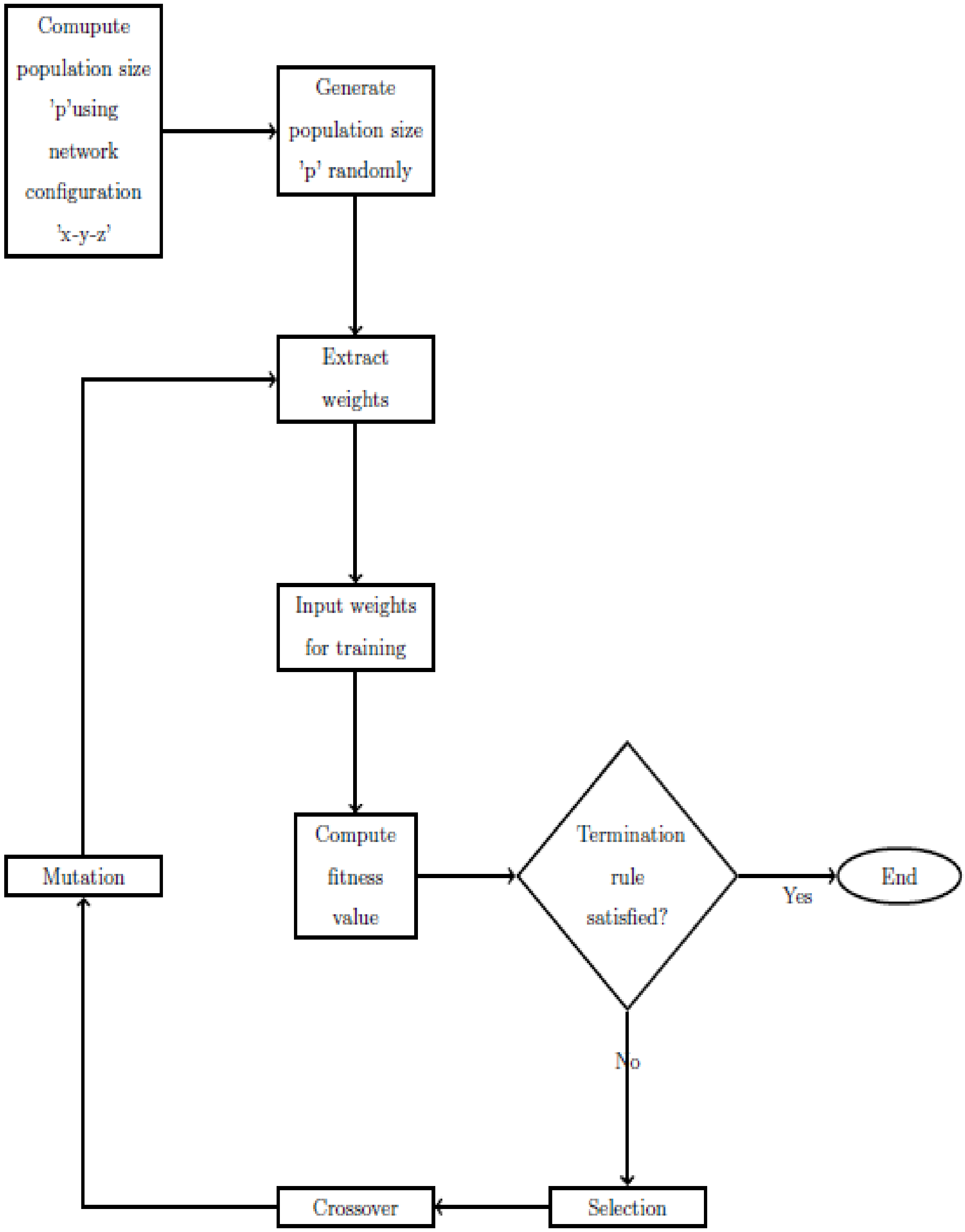
.

where N = training data.

Step 5: Fitness value for chromosome Ci using the following equation is found out as:

$$\boxed{F_i = \frac{1}{E_i} = \frac{1}{\sqrt{\frac{\sum_{k=1}^{k=N} E_k}{N}}}}$$

Fig 1.1 Flow chart for Neuro-GA Model



4. RESULT ANALYSIS:

In this proposed approach, a total population size of 100 is taken. The procedure converges when the fitness value of 90 % of chromosomes are same. The Neuro GA approach makes 300 number of iteration to have the chromosomes having similar fitness value. The result is shown in Figure 4.2 shows the plot between number of chromosome having same fitness value versus number of iteration. It can be inferred from the Figure 4.2 that as the number of iterations increase number of chromosomes having same fitness value is rising accordingly.

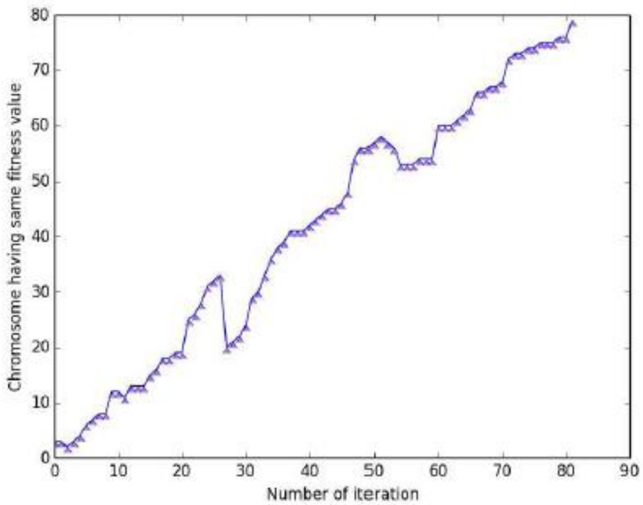


Table 1.1: Performance of Neuro GA Algorithm

Error	MAE	MARE	RMSE	SEM	Accuracy
Neuro-GA	0.063	0.089	0.085	0.037	94%

5. SUMMARY

In this paper we propose approaches like Genetic Algorithm (GA), Memetic Algorithm (MA) and Particle Swarm Optimization (PSO) for web service selection based on the non functional parameter. Accordingly the quality of service is checked. Quality of Service plays major role in the selection of web services. Quality of Service is determined by the execution of non functional parameters like availability, robustness, simplicity, reliability etc. Depending on the client‟s requirement of the service and the non functional parameters the web service should be selected. There are many approaches for the Web Service Selection as discussed in the paper. We are proposing Particle Swarm Optimization for the selection of web service as this optimization technique is very simple, it has very few parameters, and easy to implement. In this paper Hybrid, ANN approach has been used for web service selection. In this approach, ANN has been used to train the network. But in case of ANN precision of output is often limited it does not admit zero error but only minimization of least square error. GA algorithm is used to determine weight set for neural network that leads to minimization of error.

REFERENCES:

[1] Simone A. Ludwig, "Memetic Algorithm for Web Service Selection", Proceedings of the 3rd workshop on Biologically inspired algorithms for distributed systems international conference on automatic communication and computing, 2011.

[2] Shang-Chia Liu, Sung-Shun Weng, "Applying Genetic Algorithm to Select Web services Based on Workflow Quality of Service", *Journal of Electronic Commerce Research*, VOL 13, NO 2, 2012.

[3] Taher, L., El Khatib, H., "A framework and QoS matchmaking algorithm for dynamic web services selection", *Proceedings of the 2nd International Conference on Innovations in Information Technology* (IIT"05), 2005.

[4] Holland, J.H., "Adaptation in Natural and Artificial Systems", University of Michigan Press, Ann Arbor. 1975

[5] Wolpert, D., Macready, W., "No free lunch theorems for optimization", *IEEE Trans. Evol. Comput.,* vol. 1, no. 1, pp.67-82, 1997

[6] Dawkins, R,. *The Selfish Gene*, New York: Oxford Univ. Press, 1976

[7] Krasnogor, N., Smith, J., "A Tutorial for Competent Memetic Algorithms: Model, Taxonomy, and Design Issues", *IEEE Trans. On Evolutionary Computation*, vol. 9, no. 5, 2005 [8] Vazquez, M., Whitley, L., "A hybrid genetic algorithm for the quadratic assignment problem", Proceeding *on Genetic Evolution Computing Conference*. pp. 135-142, 2000

[9] Ku, K., Mak, M. "Empirical analysis of the factors that affect the Baldwin effect", *Lecture Notes in Computer Science, Parallel Problem Solving From Nature*, pp. 481-490, 1998

[10] Morris, G.M., Goodsell, D.S., Halliday, R.S., Huey, R., Hart, W.E., Belew, R.K., Olson, A.J., "Automated docking using a lamarkian genetic algorithm and an empirical binding free energy function, Journal Computing". *Chem.,* vol. 14, pp. 1639-1662, 1998

[11]M.C. Frédérique, Claudia Catalina, Evaluation "Framework for Quality of Service in Web Services: implementation in a pervasive environment", Master Thesis Master Research in Informatics Specialized in Technologies of Information & Web

[12] J. O'sullivan, D. Edmond and A.T. Hofstede., "What's in a service? Towards accurate description of non-functional service properties", *Distributed and Parallel Databases Journal*, September 2002.
[13] N.P.Padhy, *Artificial Intelligence and Intelligent Systems*, Oxford University Press, 12th impression 2013.

[14] WSMO Deliverable, D28.4 V0.1, *Non Functional Properties of Web Services* WSMO Working Draft – October 25, 2006. [15] Mona niyakan Lahiji, *Particle Swarm Optimization (PSO) Algorithm*, CNB scholar journals, Journal of Biology and today's world volume 1, issue 1, 2012

[15] S. Rajasekaran and G. V. Pai, NEURAL NETWORKS, FUZZY LOGIC AND GENETIC ALGORITHM: SYNTHESIS AND APPLICATIONS (WITH CD). PHI Learning Pvt. Ltd., 2003.