Job Scheduling Problem with Fuzzy Neural Network by using the MapReduce Model in a Cloud Environment

Forough zare
Department of Computer,
Science and Research Branch,
Islamic Azad University
Khouzestan, Iran

Mashallah Abbasi Dezfoli
Department of Computer,
Science and Research Branch,
Islamic Azad University
Khouzestan, Iran

Reza Javidan
Department of Computer
Engineering and IT, Shiraz
University of Technology,
Shiraz, Iran

ABSTRACT

Cloud computing is a solution for processing large amounts of data. Therefore, Google introduced map reduce as a programming model for large scale data applications in the cloud environment. Map reduce is used for data processing and parallel computing. The Apache Hadoop is an open source implementation of mapreduce. However job shop scheduling problem (JSSP) is an important issues that is one of the most popular NP hard, it is necessary to find a faster solution for large scale problems. For this purpose, fuzzy neural network must be use to solve this kind of optimization problem. In this paper, we proposed new novel method by using a fuzzy neural network with map reduce model to solve job shop scheduling problem, implementation and results are presented. The experiments of our proposed method are performed for well-known problem instances from job scheduling. The results show our method has high convergence speed and less execution time compared with Genetic algorithm.

Keywords

Cloud Computing, Map Reduce, Fuzzy Neural Network, Job Scheduling.

1. INTRODUCTION

Job scheduling is used to allocate particular resources for a certain tasks in particular time [1]. Job shop scheduling is usually a strongly NP-complete problem of combinatorial optimization problems and is the most typical one of the production scheduling problems. Means that the solution space will expand exponentially, and consequently the performance of traditional mathematical or meta-heuristic algorithms can hardly be satisfactory [2]. It is usually very hard to find its optimal solution [3]. So, fuzzy neural network structure is capable to solving the scheduling problem. The term Neural Network describes the adjusting weights of the hidden layer to match the input and output. No mathematical model is necessary for neural network. It's learning from examples. In neural network initially behaves as block box behavior [4]. On the other hand, Cloud Computing emerges for varieties of internet businesses, many computing frameworks are proposed for the huge data store and highly parallel computing needs [5]. Apache hadoop [6] is an open source implementation of the Google's MapReduce [7] parallel processing framework. MapReduce [8] is one of the most popular programming models designed for data centers. It was originally proposed by Google to handle large-scale web search applications and has been proved to be an effective programming model for developing data mining, machine learning and search applications in data centers. MapReduce was designed to support parallel large scale data processing on a cluster of commodity hardware, which is also known as cloud computing [9]. Map-Reduce is a scalable programming model [10]. It is abstracted such that only two main functions are written, Map and Reduce, where input and output are expressed in key-value object pairs. This model allows the programmer to simply focus on the logic through these two functions, and not worry about intra-node communication, task monitoring, or task-failure handling [10]. As well, both the map and reduce functions do not have to account for the size of data or even the underlying cluster that they are operating on. Mapper (Map Phase) - The Hadoop framework reads input data from large files and split it into smaller portions (data blocks) which are then assigned to a mapper task. The data portions are read and wrapped into a (key, value) object pair and assigned to the map function. For an input of a key and a value, the function outputs (key, value) pair(s) which are then forwarded to the Reducers [11]. Reducer (Reduce Phase) - The output from the map functions is then sorted such that object values with the same key are grouped, assigned and transmitted to a reduce function. These grouped values are then processed by the Reduce functions and written back into the file system.

Therefore, the aims of in this paper are to solving the job scheduling problem by fuzzy neural network with map reduce model that provide optimal solution in short time is presented.

This paper is organized as follows: In Section 2, we review related works briefly. In section 3, formulation of the job-shop scheduling problem is presented. In Section 4, the proposed method is given. Experimental results are presented in Section 5 and the conclusions are listed in Section 6.

2. RELATED WORK

In this section, we present some of researches that are related with this work.

Huang and Lin [9] used a Genetic Algorithm for Job Shop Scheduling Problems using MapReduce. It is shown that larger population sizes not only tend to yield better solutions, but also require fewer generations. Therefore, it is clear that when dealing with a hard problem such as JSSP, an existing GA can be improved by massively scaling up populations with MapReduce, so that the solution can be parallelized and completed in reasonable time. Venkatesa Kumar and Dinesh [4] present fuzzy neural network algorithm for job scheduling in cloud computing. The algorithm is implemented with the help of simulation tool (CloudSim) and the result obtained reduces the total turnaround time and also increase the performance. Jian-feng et al [12] represented task scheduling algorithm based on improved ant colony optimization in cloud computing environment. It is proven a better scheduling result with shorter total-task-finish time and mean-task finish time and batter load balance can be got through it in the simulation environment. Baomin et al [13] present job scheduling algorithm based on berger model in cloud environment. In the job scheduling process, the algorithm establishes dual fairness constraint. The experimental results show that the algorithm can effectively execute the user tasks and manifests better fairness. Yang and Wang [14] a new adaptive neural network and heuristics hybrid approach for job-shop scheduling is presented. Two heuristics are presented, which can be combined with the neural network. Computer simulations have shown that the proposed hybrid approach is of high speed and efficiency. The strategy for solving practical job-shop scheduling problems is provided. Tayal [15] purposed an optimized algorithm based on the Fuzzy-GA optimization which makes a scheduling decision by evaluating the entire group of task in the job queue.

3. FORMULATION OF THE JOB-SHOP SCHEDULING PROBLEM

Generally for job shop scheduling problem there are two types of constraints: Sequence constraint (SC) and resource constraint (RC). The first type states that two operations of a job cannot be processed at the same time. The second states that no more than one job can be performed on one machine at the same time. Job scheduling can be viewed as an optimization problem, bounded by both sequence and resource constraints [16].

Denote $N = \{1, \ldots., n\}$and $M = \{1, \ldots, m\}$, where n and m are the numbers of jobs and machines. Let n_i be the operation number of jobs i. O_{ikp} represents operation k of job i on machine q, S_{ikq} and T_{ikq} represent the starting time and processing time (which is known in advance) of O_{ikp}, S_{ieiq} and T_{ieiq} represent the starting time and processing time of the last operation of job i respectively. Denote r_i and d_i as the release date (earliest starting time) and due date (latest ending time) of job i. Let pi denote the set of operation pairs $[O_{ikp}, O_{ilq}]$ with precedence restriction of job i, where operation O_{ikp} must precede operation O_{ilq}. Let R_q be the set operations O_{ikp} that will be processed on machine q. The mathematical programming formulation of the considered Job-shop scheduling problem is presented as follows [16]:

$$S_{ilq} - S_{ikp} \geq T_{ikp}, \; [O_{ikp}, O_{ilq}] \; \epsilon \; P_{i\,,} \qquad K, l \epsilon \{1, \ldots, n_i\} \quad (1)$$

$$S_{jlq} - S_{ikq} \geq T_{ikq} \qquad Or \qquad S_{ikq} - S_{jlq} \geq T_{jlq}, \\ O_{ikq}, O_{jlq} \epsilon \; R_q \;, \;\; i, j \epsilon \; N, \;\; q \epsilon M \qquad (2)$$

$$r_i \leq S_{ijq} \leq d_i - T_{ijq}, \qquad i \epsilon N, \;\; j \epsilon \{1, \ldots, n_i\}, \qquad q \epsilon M \qquad (3)$$

Equation (1) represents the sequence constraint. Equation (2) in a disjunctive type represents resource constraints. Equation (3) represents the release date and due date constraints. In this study we used constraint satisfaction adaptive neural network (CSANN), based on the general neural unit, CSANN constraints three kinds of units: ST-units, SC-units and RC units. The first kind of units represent the starting times of all operations. Each ST-unit represents one operation of job shop scheduling problem with its activation representing the starting time of the operation. The second represent whether the sequence constraints are violated. The third represent whether the resource constraints are violated. The net input and activation functions of an ST-unit, ST_i are defined as (4) [16]:

$$N_{ST_i}(t) = \sum_j \left(W_{ij} \times A_{SC_j}(t)\right) + \sum_k \left(W_{ik} \times A_{RC_k}(t)\right) + \\ A_{ST_i}(t-1) \qquad (4)$$

Formulations of SC units and RC units in propose method is presented.

4. THE PROPOSED METHOD

Map Reduce model consists of basically two phases, map and reduce. According to the proposed method, some part of neural network computing is performed in map phase and another part is done in reduce phase. At first, job scheduling problem input data should be converted into (key, value) pairs that calculated in map phase of SC-unit. After the calculation of SC neural units in map phase, the intermediate (key, value) pairs produced that used reduce phase input of SC-unit. RC neural units in a new map phase received the output of reduce phase from previous phase. Similarly, after calculations in map phase of RC-unit, the intermediate (key, value) pairs produced that used as a reduce phase input of RC-unit. The (key, value) pairs that are produced in the end of reduce phase included new start time (ST) of all operations in scheduling problem. We used fuzzy due date to make decision for the completion time of each job to prevent delay between operations. This results related to first cycle and all of these steps are iterated until convergence and optimal solution be obtained. Block diagram of proposed method is shown in figure 3. Phases of the proposed method in the follow are presented.

4.1 Map Phase of SC-Units

At first, job scheduling problem input data should be converted into (key, value) pairs. We considered two matrices to perform calculations, one matrix for processing time and one matrix for machine allocation. We used i as a job, j as an operation and k as a machine. All of this index used as the key (i, j, k) and processing time as the value. Each unit of SC-block contains two ST-units, responding to two operations of a job, and one SC-unit, representing whether the sequence constraint between these two operations is satisfied [16] (see Figure 1).

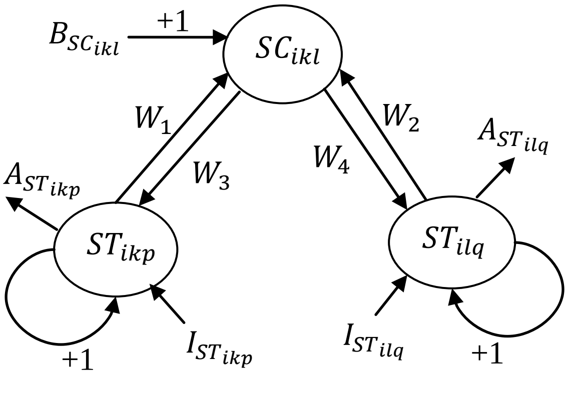

Fig 1: SC-block unit [16]

SC-units received activations of ST Units with connection weight W_1 and W_2 by equation of (5):

$$N_{C_i}(t) = \sum_j \left(W_{ij} \times A_{ST_j}(t)\right) + B_{C_i} \qquad (5)$$

Where C_i equals SC_i or RC_i , and B_{Ci} is the bias of neural unit SC_i or unit RC_i. The bias B_{Ci} is added to the incoming weighted activations of the connected ST-units ST_i's and equals the processing time of a relative operation, formulated in this equation [16]. The activation function of a SC-unit or a RC-unit is a deterministic linear-segment function, defined as follows (6):

$$A_{C_i}(t) = \begin{cases} 0\ , & N_{C_i}(t) \geq 0 \\ -N_{C_i}(t), & N_{C_i}(t) < 0 \end{cases} \tag{6}$$

The activation of a SC-unit or RC-unit being greater than zero means the corresponding sequence constraint or resource constraint is violated and there are feedback adjustments from this SC-unit or RC-unit to connected ST-units through adaptive weighted connections. The weights and bias are valued [16] as follows (7):

$$W_1 = -1, \quad W_2 = 1, \quad W_3 = -W, \quad W_4 = W, \quad B_{SC_{ikl}} = -T_{ikp} \tag{7}$$

Where W is a positive feedback adjustment parameter (the same with following equations where W appears).

After calculations of map phase, intermediate (key, value) pairs built that used as a reduce phase input of SC-units. Map Intermediate (key, value) pairs written to file system in form of $((i, j, k), A_{Ci})$. (i, j, k) as the key and Activation of SC units as the value.

4.2 Reduce Phase of SC-Units

In reduce phase, reduce function of our method selected some suitable activations for ST-Units calculations. Feedback adjustments of SC units calculated in this Phase. ST units received activations of SC Units with connection weight W_3 and W_4 by [16] equation of (8):

$$N_{ST_i}(t) = \sum_j \left(W_{ij} \times A_{SC_j}(t)\right) + A_{ST_i}(t-1) \tag{8}$$

The activation function of ST-units [16] is a deterministic linear-segmented function as follows (9):

$$A_{ST_i}(t) = \begin{cases} r_i\ , & N_{ST_i}(t) < r_i \\ N_{ST_i}(t), & r_i \leq N_{ST_i}(t) \leq d_i - T_{ST_i} \\ d_i - T_{ST_i}, & N_{ST_i}(t) > d_i - T_{ST_i} \end{cases} \tag{9}$$

Where r_i and d_i are the release date and due date of job i to which the operation, corresponding to unit ST_i , belongs. T_{ST_i} is the processing time of the operation corresponding to unit ST_i. This activation function implements the release date and due date constraints described by equation (3).

After calculations of reduce phase, output (key, value) pairs produced. Reduce output (key, value) pairs written to file system in form of $((i, j, k), A_{ST})$. (i, j, k) as the key and Activation of ST units as the value. Our method used this (key, value) pairs in new mapreduce calculations in RC units.

4.3 Map Phase of RC-Units

The calculations in this phase were similar to map phase of SC-units. Each unit of RC-block contains two ST-units, responding to two operations sharing the same machine, and one RC-unit, representing whether the resource constraint between these two operations is satisfied (see Figure 2).

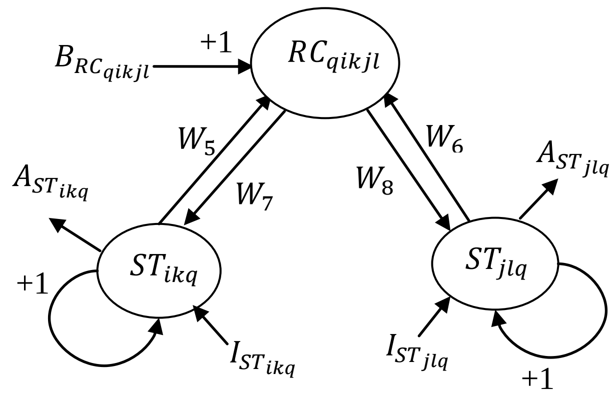

Fig 2: RC-block unit [16]

Figure 2 presents an example of RC-block unit, denoted by RCB_{qikjl} , representing the resource constraint between O_{ikq} and O_{jlq} on machine q. At time t during the processing of network, the weights and bias are adaptively valued as following two cases show [16].

RC-units received activations of ST-Units with connection weight W_5 and W_6 by equations of (10) or (11). RC block units have the same definition as (5), (6).

In this case RCB_{qikjl} represents a sequence constraint described [16] by the second disjunctive equation of equation (2).

Case1: If $S_{ikq}(t) \leq S_{jlq}(t)$, equation (10) holds

$$W_5 = -1, \quad W_6 = 1, \quad W_7 = -W, \quad W_8 = W, \quad B_{RC_{qikjl}} = -T_{ikq} \tag{10}$$

Case 2: If $S_{ikq}(t) \geq S_{jlq}(t)$, equation (11) holds

$$W_5 = 1, \quad W_6 = -1, \quad W_7 = W, \quad W_8 = -W, \quad B_{RC_{qikjl}} = -T_{jlq} \tag{11}$$

Similarly, intermediate (key, value) pairs built that used as a reduce phase input of RC-unit. Map Intermediate (key, value) pairs written to file system in form of$((i, j, k), A_{Ci}\ )$. (i, j, k) as the key and Activation of RC units as the value.

4.4 Reduce Phase of RC-Units

At the end of calculations, reduce function of RC-units selected some suitable activations for ST Unit too. Feedback adjustments of RC units calculated in this Phase. ST-units received activations of RC-units with connection weight W_7 and W_8 by equation [16] of (12):

$$N_{ST_i}(t) = \sum_k \left(W_{ik} \times A_{RC_k}(t)\right) + A_{ST_i}(t-1) \tag{12}$$

The activation function of ST-units have the same definition as (9). After calculations of this reduce phase, output (key, value) pairs produced. Reduce output (key, value) pairs written to file system in form of $((i, j, k), A_{ST})$. (i, j, k) as the key and Activation of ST units as the value. This output (key, value) pairs represented start time of all operations related to first cycle. For each cycle, membership function of fuzzy due date by equation (13) represented the satisfaction degree of completion time of job. In mathematics, membership function $fi\ (Ci)$ of job i can be denoted by following formula [17]:

$$f_i(C_i) = \begin{cases} 1 & C_i \le d_i^L \\ \frac{d_i^U - C_i}{d_i^U - d_i^L} & d_i^L \; < C_i < d_i^U \\ 0 & C_i \ge d_i^L \end{cases} \qquad (13)$$

Where C_i is the completion time of job i. We can see that the membership function is specified by d_i^l and d_i^u $(i = 1, \ldots, n)$. d_i^l can be viewed as the earliest due date for job i, and d_i^u is the latest due date [17].

5. EXPERIMENTAL RESULTS

In this study, we used different number of jobs to examine our proposed method. We selected various instances from job scheduling problem like FT3, FT4×3, FT6, FT10, FT20 [18], La40 [19] and Swv14 [20]. The performance of our method is shown in Table 1. Figure 4 shows execution time vs. job size. Our proposed method has been compared with Genetic algorithm [9]. The comparisons of execution time for these methods are present in table 2. Figure 5 shows the result of proposed method and the result of Genetic algorithm for La40 instance.

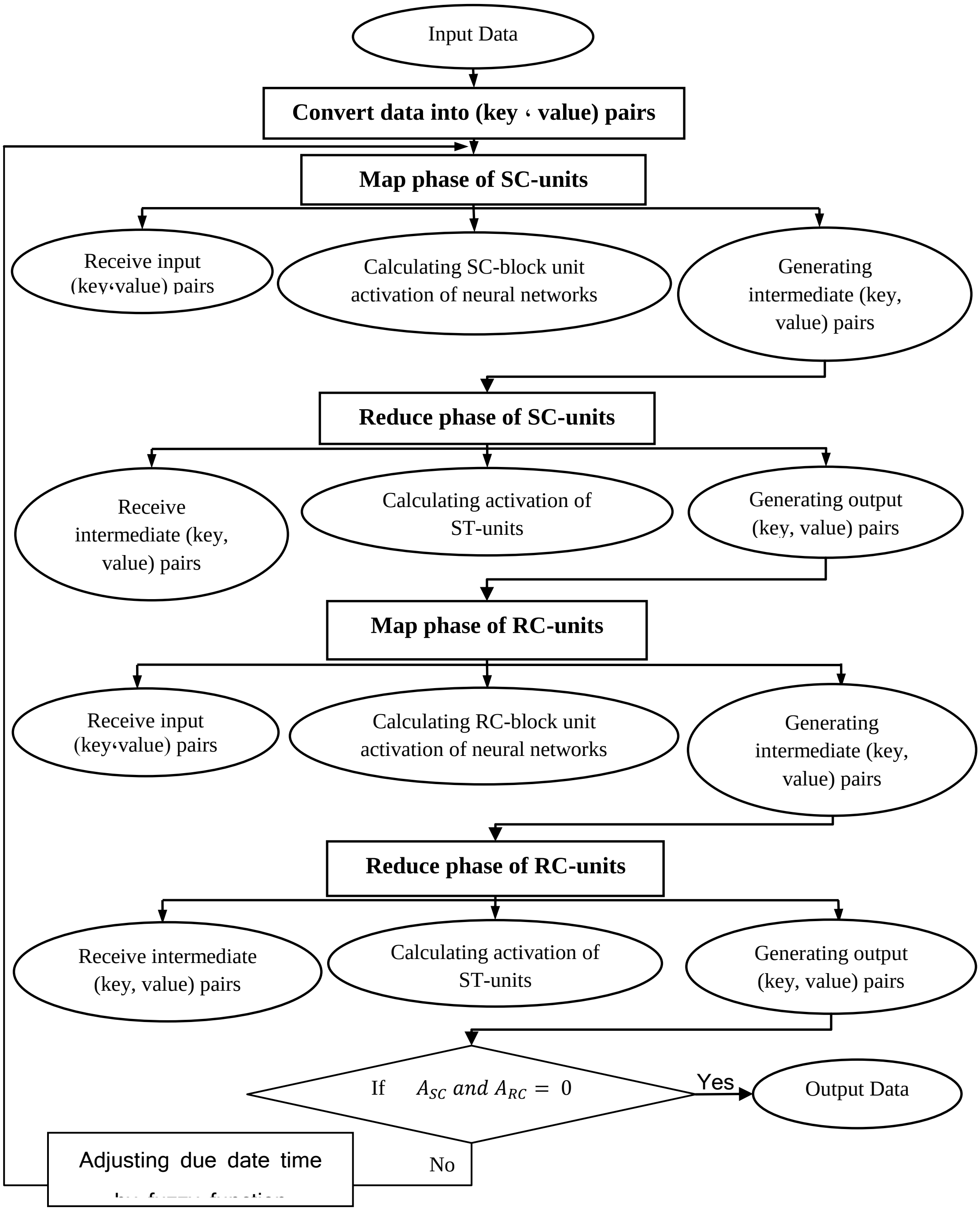
Input Data
Convert data into (key ‹ value) pairs
Map phase of SC-units
Receive input (key‹value) pairs
Calculating SC-block unit activation of neural networks
Generating intermediate (key, value) pairs
Reduce phase of SC-units
Receive intermediate (key, value) pairs
Calculating activation of ST-units
Generating output (key, value) pairs
Map phase of RC-units
Receive input (key‹value) pairs
Calculating RC-block unit activation of neural networks
Generating intermediate (key, value) pairs
Reduce phase of RC-units
Receive intermediate (key, value) pairs
Calculating activation of ST-units
Generating output (key, value) pairs
If A_{SC} and A_{RC} = 0
Yes
Output Data
No
Adjusting due date time by fuzzy function

Fig 3: Diagram of proposed method

Table 1: Execution time for number of jobs on different machines

Problem name	Number of jobs	Number of machine	Execution time (Sec)
FT3	3	3	2.54
FT4×3	4	3	1.42
FT6	6	6	8.46
FT10	10	10	32.75
FT20	20	5	120.43
LA40	15	15	40.54
SWV14	50	10	580.89

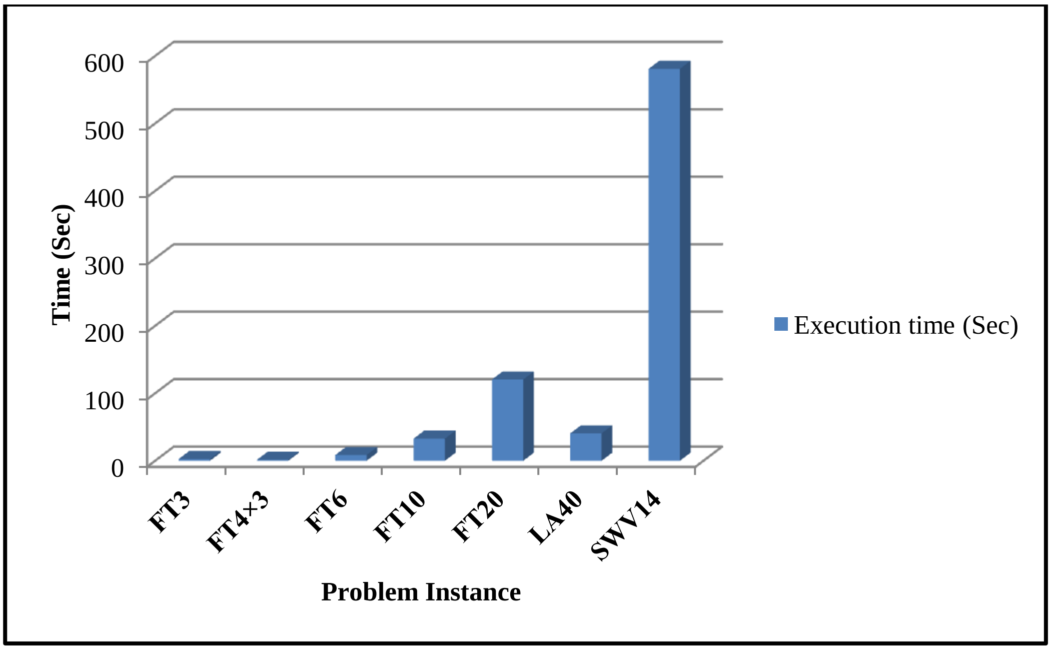

Fig 4: Flow execution time proposed method for various problems

Table 2: Compared proposed method with Genetic algorithm for LA40

Method	Number of jobs	Number of machine	Execution time (Sec)
Fuzzy neural network with map reduce model	15	15	40.54
Genetic algorithm with map reduce model	15	15	356.067

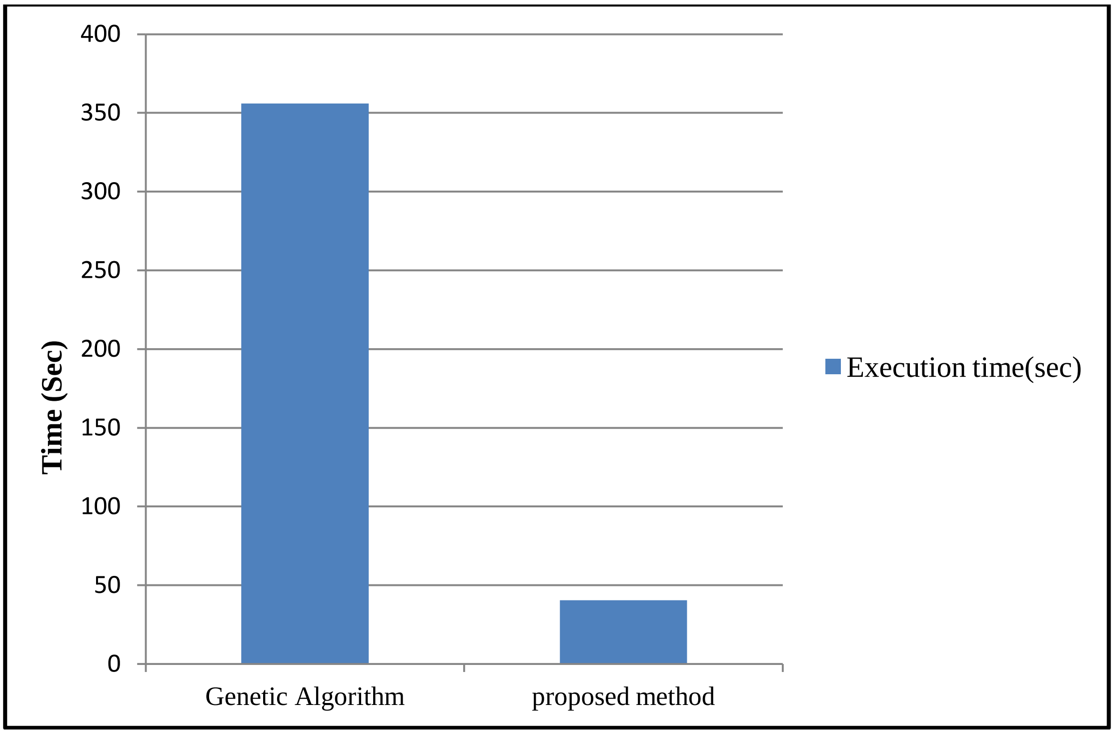

Fig 5: Comparative performance proposed method and Genetic algorithm

6. CONCLUSION

In this paper, a scheduling problem based on fuzzy neural network using the map reduce model is proposed for job scheduling problem in cloud environment. We modified conventional fuzzy neural network to implement in map reduce model. For this propose we split neural network units and have done the calculations separately in map and reduce phases. We used various instance like FT10, FT20, Swv14 and La40 to investigate performance of our method. The results indicated that parallel solubility and reduced calculations using the map reduce model provide to reach optimal solution in very short time. In comparison with genetic algorithm, our proposed method has a better convergence speed and less execution time.

7. REFERENCES

[1] Zhao, C. Zhang, S. Liu, Q. Xie, J. and Hu, J. 2009. Independent Tasks Scheduling Based on Genetic Algorithm in Cloud Computing, IEEE paper 2009, Pp. 978-1-4244-3693-4.

[2] Zhang, R. and Wu, C. 2008. bottleneck machine identification based on optimization for the job shop scheduling problem, ICIC Express Letters ICIC International , ISSN 1881-803X Volume 2, Number 2, pp. 175—180.

[3] Yang, Sh. Wang, D. Chai, T. and Kendall, G. 2009. An improved constraint satisfaction adaptive neural network for job-shop scheduling, Springer, DOI 10.1007/s10951-009-0106-z.pp.

[4] Venkatesa Kumar V. and Dinesh K. 2012. Job Scheduling Using Fuzzy Neural Network Algorithm in Cloud Environment, Bonfring International Journal of Man Machine Interface, Vol. 2, No. 1, ISSN 2277 – 5064.

[5] Dean J. and Ghemawat. S. 2004. MapReduce: simplified data processing on large clusters, Sixth Symposium on Operating System Design and Implementation (OSDI’04), pp. 1-13.

[6] Apache Hadoop. http://hadoop.apache.org.

[7] Dean J. and Ghemawat. S. 2004. Mapreduce: Simplified data processing on large clusters. OSDI ’04, pages 137–150.

[8] Hadoop MapReduce, hadoop.apache.org/mapreduce.

[9] Huang Di-Wei. and Lin J. 2010. Scaling Populations of a Genetic Algorithm for Job Shop Scheduling Problems using MapReduce, This work was supported in part by the NSF under awards IIS-0836560 and IIS-0916043, and also in part by Google and IBM, via the Academic Cloud Computing Initiative (ACCI).

[10] White, Tom.2009. Hadoop: The Definitive Guide. O'Reilly.

[11] MapReduce. http://developer.yahoo.com/hadoop/tutorial/module4.html.

[12] Jian-feng, LI. Jian, P. Cao, X and Hong-you LI. 2011. A Task Scheduling Algorithm Based on Improved Ant Colony Optimization in Cloud Computing Environment, Energy Procedia 13 (2011) 6833 – 6840

[13] Xu, B., Zhao C., Hu. E, Hu. B, “Job Scheduling algorithm using berger model in cloud environment”, Elsevier in Advances in Engineering Software, 2011, Vol. 42 , No. 7, Pp. 419-425.

[14] Yang, Sh. and Wang, D. 2008. A New Adaptive Neural Network and Heuristics Hybrid Approach for Job-Shop Scheduling, This research was supported by the National Nature Science Foundation (No. 69684005) and National

High -Tech Program of P. R. China (No. 863-511-9609-003) and was done when Shengxiang Yang was pursuing his Ph. D. degree.

[15] Tayal, S. 2011. Tasks Scheduling optimization for the Cloud Computing Systems. IJAEST International Journal of Advanced Enginnering Sciences And Technologies Vol No.5, Issue No.2, 111-115.

[16] Yang, Sh. Wang, D. Chai, T. and Kendall, G. 2009. An improved constraint satisfaction adaptive neural network for job-shop scheduling, Springer, DOI 10.1007/s10951-009-0106-z.pp.

[17] Xie, Y. Xie, J. and Li, J. 2005. Fuzzy Due Dates Job Shop Scheduling Problem Based on Neural Network, Department of Automation, Shanghai Jiaotong University, Postbox 280, Shanghai 200030, China. Publisher Springer-Verlag Berlin, Heidelberg, ISBN: 3-540-25912-0 978-3-540-25912-1. 782-787.

[18] Fisher, H. and Thompson, G. L. 1963. Probabilistic learning combinations of local job-shop scheduling rules. In J. F. Muth & G. L.Thompson (Eds.), Industrial scheduling (pp. 225–251). Englewood Cliffs, New Jersy: Prentice Hall.

[19] Lawrence, S. 1984. Resource constrained project scheduling: An experimental investigation of heuristic scheduling techniques (supplement). Pittsburgh, PA: Graduate School of Industrial Administration, Carnegie-Mellon University.

[20] Storer R.H., Wu S.D., Vaccari. R. 1992. New search spaces for sequencing instances with application to job shop scheduling, Management Science 38, 1495-1509.