Optimisation of Parallel Machine Scheduling Using Neuro-Fuzzy Approach

A Project Report

Submitted by

D.Kavitha - 71204409004

in partial fulfillment for the award of the degree
of

Master of Engineering
in
Industrial Engineering

DEPARTMENT OF MECHANICAL ENGINEERING
KUMARAGURU COLLEGE OF TECHNOLOGY
COIMBATORE – 641 006

ANNA UNIVERSITY :: CHENNAI 600 025

APRIL – 2006

ANNA UNIVERSITY:: CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report entitled **"Optimisation of Parallel Machine Scheduling Using Neuro-Fuzzy Approach "** is the bonafide work of

D.Kavitha - Register No. 71204409004

Who carried out the project work under my supervision.

Signature of the Head of the Department	Signature of the supervisor
Dr. T. P. MANI	Mr. A.RAJESH
HEAD OF THE DEPARTMENT	LECTURER

30/06/06

Internal Examiner

Dr. T.P. Mani
B.E., M.E. Ph.D DML., MIE MNIOR., MISTE.,
Dean & HoD / Dept of Mech. Engg.
Kumaraguru College of Technology
Coimbatore - 641 006

External Examiner

S. R. Devadasan, Ph.D.
Professor
Production Engineering Department
PSG College of Technology
Coimbatore - 641 004, INDIA

DEPARTMENT OF MECHANICAL ENGINEERING
KUMARAGURU COLLEGE OF TECHNOLOGY
COIMBATORE 641 006

Kumaraguru College of Technology

Coimbatore - 641 006

Jointly Organized by:

Department of Mechanical Engineering & Industry Institute Partnership Cell

CERTIFICATE OF MERIT

This is to certify that Mr. / Ms. D . KAVITHA , P. G. SCHOLAR

KUMARAGURU COLLEGE OF TECHNOLOGY , COIMBATORE *has*

*ticipated in the National Conference on **"The Impact of Information*

chnology on Manufacturing" **(NCIITM – 2006)**, *on 24th and 25th Mar'06*

presented a paper entitled OPTIMIZATION OF PARALLEL MACHINE SCHEDULING

USING NEURO - FUZZY APPROACH

(Prof. C. R. Kamalakannan)
zing Secretary - NCIITM - 2006,
AP/ME

(Dr. T. P. Mani)
Dean & Head/Mech.

(Dr. K. K. Padmanabhan)
Principal

THE GANDHIGRAM RURAL INSTITUTE – DEEMED UNIVERSITY

(Accredited with FIVE star status by NAAC)

GANDHIGRAM –624 302

DEPARTMENT OF MATHEMATICS

UGC - SAP (DRS) NATIONAL CONFERENCE ON INTELLIGENT OPTIMIZATION MODELING

(NCIOM - 2006)

This is to certify that Dr./~~Shri.~~/Ms. D. KAVITHA, STUDENT, M.E (INDUSTRIAL ENGINEERING) *of* KCT, COIMBATORE.

has participated in the National Conference on Intelligent Optimization Modeling held during 03-04 March, 2006 and presented a paper entitled "OPTIMIZATION OF PARALLEL MACHINE SCHEDULING USING NEURAL NETWORK TECHNIQUE"

Co-authored by A. RAJESH, Dr. V. SELLADURAI.

andhigram

4th March 2006

Dr. R. UTHAYAKUMAR

Convenor

ABSTRACT

The present globalized environment has brought a tough competition among manufacturers. The jobs are to be manufactured in varieties and in different quantities with the available resources. The manufacturer has to focus on balancing the work among the available machines and at the same time the completion time of the jobs has to be reduced in order to deliver the products on the due date.

Workflow balancing on a shop floor helps to remove bottlenecks present in the manufacturing system. Workflow refers to the total time during which the work centers are busy. Earlier researchers have also specified the method for jobs to be executed in parallel in order to balance the workflow to each machine.In this project work the Neuro-Fuzzy approach has been adopted. In parallel machine scheduling there are 'm' machines to which 'n' jobs are to be assigned based on different priority strategies. The procedure is based on the idea of balancing the workload among machines. Five different priority strategies are followed for the selection of jobs namely RANDOM, SPT, LPT, FCFS, LCFS for the selection of jobs for workflow balancing. The Relative Percentage of Imbalance (RPI) is adopted among the parallel machines to evaluate the performance of these strategies in a standard manufacturing environment using Neuro-Fuzzy approach.

The practical benefit of neural network approach is that it incrementally learns the sequencing knowledge and can apply the knowledge for sequencing a set of jobs on a real time basis. The fuzzy logic approach is used to train the uncertain weights which are not trained by using neural network The Neural Network approach further improves solution quality by combining with fuzzy logic. A validation has been done by comparing the results of Neuro-Fuzzy approach with the heuristic approach. In Neuro-Fuzzy approach, SPT strategy gives optimized results when compared to other strategies. The Neuro-Fuzzy approach is quite effective and efficient for selecting the

ஆய்வு சுருக்கம்

இக்காலகட்டத்தில் உற்பத்தி தொழிற்சாலைகளுக்கு இடையே உள்ள போட்டியை தவிர்க்க காலமுறைப்படுத்துதல் மிகவும் அவசியமான ஒன்றாகும். ஒரு தொழிற்சாலையை மேம்படுத்த காலமுறைப்படுத்துதலை சார்ந்த செயல்திறன் அளவீடு நமது தேவைக்கேற்ப உயர்த்த வேண்டும். இந்த ஆராய்ச்சியில் காலமுறைப்படுத்துதலின் செயல்திறன் அளவீடுகளை கண்டறிய "நீயுரல் நெட்வொர்க் மற்றும் பஜி லாஜிக்" தொழில் நுட்பம் பயன்படுத்தப்படுகிறது.

இந்த காலமுறைப்படுத்துதலில் பல பணிகள் மற்றும் பல இயந்திரங்கள் உள்ளன. அதில் எந்தப்பணியை எந்த இயந்திரத்தில் செய்ய வேண்டும் என்பதனையும், எவ்வளவு சீக்கிரத்தில் அந்த பணியை முடிக்க வேண்டும் என்பதற்கும் பல வரிசை முறை கண்டறியப்பட்டுள்ளது.

அவற்றுள் சிறந்த வரிசையை கண்டறிவதே இந்த கட்டுரையின் முக்கிய நோக்கம். சிறந்த வரிசையை கண்டறிய மூன்று முக்கிய காரணிகள் உள்ளன. அவைகள் (1) பணி நிறைவடையும் நேரம் மிக குறைவாக இருக்க வேண்டும் (2) பணிகள் காலதாமதமாவதை குறைக்க வேண்டும் (3)இயந்திரங்கள் முழுமையாக பயன்படுத்தப்படவேண்டும்.

இறுதியாக இந்த சிறந்த வரிசையை கண்டறிய " நீயுரல் நெட்வொர்க் மற்றும் பஜி லாஜிக்" என்னும் தொழில் நுட்பம் பயன்படுத்தபடுகிறது. இதன் மூலம் மிகவும் எளிதாகவும் துல்லியமாகவும் சிறந்த வரிசையை கண்டறியலாம்.

ACKNOWLEDGEMENT

The author expresses her sincere appreciation to her respected Project supervisor **Mr.A.Rajesh,** Lecturer, Department of Mechanical Engineering, Kumaraguru College of Technology Coimbatore, and the faculty members whose valuable guidance, inspiration and continuous encouragement throughout the course made it possible to complete this project work well in advance.

The author would like to thank **Dr.T.P.Mani,** Head of the Department, Department of Mechanical Engineering, Kumaraguru College of Technology, Coimbatore for his motivation in completing this project.

At this colossal success of this project, the author take immense pleasure to acknowledge with infinite thanks to our beloved Principal, **Dr.K.K.Padmanaban,** Kumaraguru College of Technology, Coimbatore who aided in eliciting this project triumphantly.

The author would like to thank all the faculties and lab technicians for the encouragement and moral support to complete this project work in a great success.

At this gracious moment, the author would like to thank her parents for their love ,affection and financial support for my education.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS & ABBREVIATIONS

n	-	No .of .Jobs
m	-	No. of Machines
P_i	-	Processing time of Jobs
X_1	-	Activation of Neuron 1
X_2	-	Activation of Neuron 2
Y	-	Output Neuron
W_1	-	Weight connection of Neuron 1 to Output
W_2	-	Weight connection of Neuron 2 to Output
B	-	Bias
W_i	-	Weight
Net	-	Input
Θ	-	Threshold
W_{ji}	-	Weight on the connection for i^{th} to j^{th} unit
E	-	Error Function
Z_i	-	Activation function
T_K	-	Target
C_I	-	Fuzzy Criterion
SPT	-	Shortest Processing Time
LPT	-	Longest Processing Time
FCFS	-	First Come First Serve
LCFS	-	Last Come First Serve
W_K	-	Workload on the Machine
W_{Max}	-	Maximum workload on the machine
RPI	-	Relative Percentage of Imbalance.
MFT	-	Mean Flow Time

CHAPTER 1

INTRODUCTION

One of the notable advances in manufacturing is the successful use of design and processing similarities of products to form product families and processing such families in exclusive machine groups. This approach, known as group technology, has brought about benefits of economics of scale, usually possible only with high volume production, to small volume batch type manufacturing systems. Production schedules for resources shared by such product families are characterized by family setup tasks that are required when processing switches from a job of one family to a job of another family. Traditionally, in such production scheduling situations, we trade-off between the number of setups and meeting delivery time line for finished products and therefore we seek the schedule that minimizes a non-decreasing function of completion times. Such cost functions, for example mean flow time, mean tardiness, number of tardy jobs and so on, are known as *regular performance measures* of a production schedule. Another notable feature of modern manufacturing is the extensive use of the Just-in-Time concept in inventory/production management where delivering goods earlier than the due dates is considered as undesirable as delivering goods late. Scheduling of such manufacturing systems thus calls for minimizing cost functions such as total *earliness and tardiness* (E/T) costs. This is an example of a *non-regular performance measure* as it is a function that decreases with increasing time first until the due date and then increases with increasing time.

1.1 PARALLEL MACHINE SCHEDULING

Jobs-to-machine assignment is based on several factors, such as precedence, availability, production rate, machine suitability (processing plan) and workload balancing with reference to the objective function. Researchers have addressed parallel machine scheduling problems with many variant approaches. More objective functions are considered to obtain an efficient schedule for

The number of machines is denoted by m and number of jobs by n. Each job is indicated by its processing time p_j where $j = 1, 2, 3 \ldots n$.

The neural network produces an efficient schedule based on an objective such as minimization of makespan time. The performance of the various strategies is measured by various measures relative to objective function.

1.2 WORKFLOW BALANCING IN PARALLEL MACHINE SCHEDULING

There are n jobs to be processed by m machines. Any machine can process at most one job at a time. No preemption is allowed. The basic objective of workload balancing is to achieve efficient utilization of productive capacity. The concept of balancing is to evenly distribute the jobs among the machine. To evaluate the different workflow balancing strategies, the authors used the performance measure of relative percentage of imbalance (RPI).

The workflow in each machine is calculated before assigning the jobs. The machine with less workflow is selected for assigning a new job from the list of jobs. The jobs are allocated to machines statically as a part of production plan. Evaluation of parallel job schedulers is based on a performance measure called workload balancing. Workload balancing aims to provide equal distribution of load on parallel machines and help to reduce the total elapsed time of jobs. The aim of the scheduling process is to increase the utilization of machines and throughput.

CHAPTER 2

LITERATURE SURVEY

Viharos et al., (2001) introduced a block-oriented framework for modeling and optimization of process chains and its applicability is shown by the results of the optimization of cutting processes. Also illustrates how the framework can support the simulation based optimization of whole production plants. The concept of hybrid Al, ML (machine learning) and simulation supported optimization of production plants was also out lines. Some results of an industrial project demonstrated the applicability of the concept where the task was to optimize the size of the ordered raw material at a plant producing one and multi layered printed wires.

Park et al., (2000) investigated the problem of scheduling jobs on identical parallel machines and proposed an extension of the ATCS (apparent tardiness cost with setups) rule developed by Lee et al which utilize some look-ahead parameters for calculating the priority index of each job

Akyol (2004) deals with the use of artificial neural networks (ANNs) to model six different heuristic algorithm applied to n job, m machine real flow shop scheduling problem with the objective of minimizing the makespan. The objective is to obtain six ANN models to be used for the prediction of the completion times for each job processed on each machine and to introduce the fuzziness of scheduling information into flow shop scheduling. Fuzzy membership function are generated for completion of the job,Waiting and machine idle times. Different methods are proposed to obtain the fuzzy parameter. To model the functional relation between the input and output variable, multi layered feed forward network is trained with error back propagation learning rule are used .

The network is trained adequately to provide an outcome (solution) faster than conventional iterative methods by its generalizing property. The results obtained from the study can extended to solve the scheduling problem in the area of manufacturing.

Salema et al., (2000) developed branch and bound [B&B] algorithm to solve the problem of minimizing the maximum completion time C_{max} on unrelated parallel machines with machine eligibility restriction when job preemption is not allowed. A customized lower bound, a search and a branching strategy are developed B&B. A machine eligibility factor is also introduced to represent the percentage of jobs eligible across all machines. Multiple instances of the problem wit different machine, job, and eligibility factor configuration are generated and solved. The performance of the B&B, which is expressed by the average number of nodes examined, was tested for different problem configurations. It was denoted that the B&B's performance improved significantly as the eligibility factor decreases below 0.5.

Shengxiang et al., (1998) a new adaptive Neural network and heuristics hybrid approach for job - shop scheduling s is presented . The neural network has the property of adapting its connection weights and biases of neural units while solving the feasible solution. Two heuristics are presented which can be combined with the neural network. One heuristic is used to accelerate the solving process of the neural network and guarantee its convergence , the other heuristic is used to obtain non delay schedules from the feasible solutions gained by 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.

Tuma.A et al., (1994) in industrial production process , materials and different forms of energy are provided, transformed respectively converted, stored and transported . With this process joint products in different stages of aggregation are emitted. Environmental impacts can be identified at any stage of the energy and material flow process. Due to the fact that production units and process are interconnected with energy and , material flows ,it is of special interest to develop production control mechanisms which control the energy and material streams in a way that utilizes available resources most efficiently and reduces emissions and by products caused by the production process . These production

The development of production control strategies depends especially on the structure of integrated production systems . If it is possible to influence the energy and material flows by the selection of special production process and an adequate allocation of jobs and aggregates, the construction of production control strategies can be reduced to a combined scheduling and technology selection problem.

Methodical production control strategies can be based on optimal algorithms (eg dynamic programming) heuristics (eg rule based approaches) and methods of machine learning (eg neural networks) due to the complexity of real production systems , it is advisable to use rule - based approaches or neural networks depending on the structure of the available production knowledge

David .E. Moriarty et al., (1994) a new method for developing good value ordering strategies in constraint satisfaction search is presented . Using an evolutionary technique called SANE, in which individual neurons evolve to cooperate and form a neural network ,problem - specific knowledge can be discovered that results in better value ordering decisions than those based on problem - general heuristics . A neural networks was evolved in a chronological backtrack search to decide the ordering of care in a resource - limited assembly line . The network required 1/30 of the back tracks of random ordering and 1/3 of the back tracks of the maximation of future options heuristic. The SANE approach should extend well to other domains where heuristic information is very difficult to discover.

S.C. Wang el al., (1996) artificial Neural Network (ANN) based expert system in manufacturing control is presented as a new approach used for discrete system. Process planning control and process scheduling control are discussed in detail by multi layer ANN . The weights series of W_{ij} 'I' the layers number, "j" the neurons number are got initially by learning samples through mapping function and then revised further by competitive learning . The mechanisms of competitive learning includes two steps : response and competition , which of only worked among these excited neurons by means of learning rules to

The learning rules involve the weights calculation by iterating and the s convergence by balance criterion . The expected values in ANN learning system in process planning are obtained by sample learning with mark giving experiment . But in scheduling control the quantities items in long term scheduling strategy are used for expected values.

Hasan Kurtaran et al., (2005) in this study optimum values of process parameters in injection molding of a bus ceiling lamp base to achieve minimum warp age are determined. Mold temperature , melt temperature , Packing pressure, packing pressure time and cooling time are considered as process parameters. In finding optimum values , advantages of element software, Mold flow, statistical design of experiments , artificial neural network and genetic algorithm are exploited. Finite element analyses are conducted for combination of process parameters designed using statistical three – level full factorial experimental design. A predictive model for war page is created using feed forward artificial neural network exploiting finite element analysis results . neural network model is validated for predictive capability and then interfaced with an effective the initial model of the bus ceiling lamp base by 46.5 %.

Christian Bierwirth et al., (1999) a general model for job shop scheduling is described which applies to static dynamic and non deterministic production environments . Next, a genetic algorithms is presented which solves the job shop scheduling problem. This algorithm is tested in a dynamic environment under different workload situations. Thereby, a highly efficient decoding procedure is proposed which strongly improves the quality of schedules. Finally this technique is tested for scheduling and rescheduling in a non - deterministic environment. It is shown by experiment that conventional methods of production control methods of production control are clearly out performed are reasonable runtime costs .

Kasin Oey et al., (2001) considers a complex jobs problem with reentrant flow and batch processing machines . A modified shifting bottle neck heuristic. (MSB) is considered for generating machine schedules to minimize the total weighted tardiness. We observe that the MSB could produce infeasible schedules where cyclic schedules are found. A cycle elimination procedure is proposed to remove the possibility of the MSB generating cyclic schedules in the solution.

Kranth kumar et al., (2000) study of neurons have always fascinated and puzzled humans. We in this following paper have discussed the basics of fuzzy logic and artificial neural network. Then we have applied the idea of ANN to the application of interfacing between a data glove and a speech synthesizer . This arrangement is used to convert hand gestures into speech . A data glove connects to a speech synthesizer through five different neural networks . The neural networks are trained using back prop algorithm . After training the system is ready to be used and can convert hand gestures to speech . This system has a vocabulary of 203 root words and also gives control over stress and speed of speaking .

Many aspects of driving are fuzzily defined and hence can be benefited by the use of Fuzzy logic . We apply this idea of fuzzy logic in controlling speed and gap between cars in a platoon within a longitudinal measure . The concept of controlling is defined in a fuzzy fashion by using additive fuzzy throttle controller .

Novel and flexible fuzzy software has been designed and implements which can be used to control the speed and gap between the cars.

Today Artificial neural network and fuzzy logic are becoming very popular in various foiled of engineering applications . ANN is becoming one of the most successful tools for training of different jobs among different machines . One of such application is of interfacing a Data – glove and a speech synthesizer this system is trained to recognize different hand gestures of hand and then convert them into speech using a speech synthesizer . This hand gestures are derived from American sign language

After training this network it can adapt to any new user after giving some sample test data by the user.

Jurgen sauer et al., (1999) the objective of multi - site scheduling is to support the scheduling activities of a global scheduler and schedulers in distributed production plants in a cooperative way . A global schedule generated on a global level must be translated into detailed schedules as part of the local scheduling process. In case of disturbance , feed back between the local and global levels is essential.

Global level data are normally aggregated, imprecise, or estimated. Previous methods focused on local production sits, in most cases without coordination . In this work we present an approach that considers the adequate modeling and processing of imprecise data for global level scheduling with in a multi site scheduling system based on fuzzy concepts . One of the goals is to create a robust prescription for the local scheduling systems which helps to reduce the effort of coordinating and rescheduling.

Pramot srinoi et al., (2004) presents a research project undertaken are industrial Research institute , Swinburne (IRIS) in the area of fuzzy scheduling . The researches commenced in march 2001 and the full project is expected to be completed in February 2004. The fuzzy based scheduling model ,in this paper ,will only deal with the part routing problem . The model will select the best alternative route with multi-criteria scheduling through an approach based in a fuzzy logic. This model applicable to the scheduling of a flexible manufacturing cell (FMC) and also a multi machine flexible manufacturing system (MMFMS).

Tommi Johtela et al., (1994) the scheduling problem of a surface mounting machine for printed circuit boards (PCB) is studied. The scheduling requires the forming of products groups . All PCBs in a group use the same component setup and can be printed sequentially without a delay. Fuzzy multiple criteria optimization is used to model the various ,partly conflicting aspects of the grouping problem.

Dominic et al., (2002) in this paper job shop model problem is scheduled with the help of gigle and Thompson Algorithm using Priority Dispatching Rule. Job Shop Scheduling (JSS) problems consist of a set of machines and a collection of jobs to be scheduled. We first reviewed the literature on job shop scheduling using search algorithms.

Scheduling in the job shop is an important aspect of a shop floor management system, which has a significant impact the performance of the shop floor. The job shop-scheduling problem is among the hardest combinational optimization problems. Each Job consists of several operations with specified processing order. Conflict based Priority Dispatching Rule (PDR) applied to schedule the job shop model by using Genetic Algorithms (GA) method.

The same job shop model is also scheduled based on operation based using Simulated Annealing (SA) and Hybrid Simulated Annealing (HAS). All the three algorithms are considered to measure the percentage utilization of machines of the job model as objective for this study and compared and results are presented. For comparative evaluation, a wide variety of data sets are used. These three methods are considered as different treatments to each problem. For each algorithm, three replications are taken for each observation and the mean value taken for significance test by using Tow-way Analysis of Variance (ANOVA) is applied to test its significance. The result are interesting, it is inferred that the percentage utilization of machines in maximum in the most of the cases under Genetic Algorithm when compared with other two algorithms like simulated Annealing and Hybrid Simulated Annealing Algorithms.

CHAPTER 3

PROBLEM DEFINITION

In the scheduling problem found in an industrial application, there are often more then one criterion mentioned that should be optimized ,e.g., prefer high priority product , minimize costs for stock etc.

Scheduling problem by the following components:

A set of orders to manufacture product which are to be scheduled Subject to several constraint, e.g. due dates, quantity, user priorities.

A set of products with information about (eventually alternative) variants (also called routing or process plans), operation with in the variant, durations of operation raw material used machine that may be used alternatively within the operation, etc.

A set of resources with different functional capabilities e.g. machine with given capacity personnel ,raw material.

A set of hard constraint that should be fulfilled but may be relaxed e.g meeting due dates, use preferred machine. Most of the objective of scheduling can be formulated by means of soft constraint. Depending on the scheduling problem, some constraint can be defined as hard or as soft constraint e.g. meeting of due dated may be a hard or a soft constraint.

One or more objective function are used to evaluate the scheduling in order to find the "best" or an optimal schedule. Hence we are arriving this optimum schedule with the objective of minimization of makespan, maximization of machine utilization and minimizing the relative percentage of imbalance.

CHAPTER 4

SOLUTION METHODOLOGY

4.1 INTRODUCTION TO ANN

Artificial neural networks are non linear information (Signal) processing devices, which are built from interconnected elementary processing devices called neurons.

An artificial neural network (ANN) is an information – processing paradigm that is inspired by the way biological nervous systems , such as the brain , process information . The key element of this paradigm is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing elements (neurons) working in union to solve specific problems . ANN like people, learn by example. An ANN is configured for a specific application , such as pattern recognition or data classification , through a learning process. Learning in biological systems involves adjustments to the synaptic connections that exist between the neurons . This is true for ANN as well.

ANN are a type of artificial intelligence that attempts to imitate the way a human brain works. Rather than using a digital model , in which all computations manipulate zeros and ones , a neural network works by creating connections between processing elements , the computer equivalent of neuron. The organization and weights of the connections determine the out put.

A neural network is a massively parallel – distributed processor that has a natural propensity for storing experimental knowledge and making it available for use . It resembles the brain in two respects. Knowledge is acquired by the network through a learning process ,and inter-neuron connection strengths known as synaptic weights are used to store the knowledge.

Neural networks can also be defined as parameterized computational nonlinear algorithms for (numerical) data / signal / image processing . These algorithms are either implemented on a general – purpose computer or are built

Artificial neural networks thus is an information – processing system . In this information - processing system , the elements called as neurons , process the informations. The signals are transmitted by means of connection links . The links possess an associated weight , which is multiplied along with the incoming signal (net input) for any typical neural network. The output signal is obtained by applying activations to the network input.

An artificial neuron is characterized by:

- Architecture (connection between neurons)
- Training or learning (determining weights on the connections).
- Activations function

All these are discussed in detail in the forthcoming subsections :

It shows a simple artificial neural network with two input neurons (X_1 , X_2) and one output neuron (y) . The inter connected weights are given by W_1 and W_2 . An artificial neuron is a p- input signal processing element ,which can be thought of as a simple model of a non branching biological neuron . Various inputs to the network are represented by the mathematical symbol , x(n) . Each of these inputs are multiplied by a connection weight . These weights are represents by w(n) . In these simplest case , these products are simply summed , fed through a transfer function to generate a result , and then delivered as output . This process lends itself to physical implementation on a large scale in a small package. This electronic implementation is still possible with other network structures , which utilize different summing functions as well as different transfer functions.

The long course of evaluation has given the human brain many desirable characteristics not present in Von Neumann or modern parallel computers .
These includes.

- Massive parallelism
- Distributed representation and computation
- Learning ability
- Generalization ability

- Inherent contextual information processing
- Fault tolerance , and
- Low energy consumption

It is hoped that devices based on biological neural networks will posses some of these desirable characteristics . Modern digital computers outperform humans in the domain of numeric computation are related symbol manipulation . However , humans can effortlessly solve complex perceptual problems (like recognizing a man in a crowd form a mere glimpse of his face) at such a high speed an extent as to dwarf the worlds fastest computer . Why is there such a remarkable difference in their performance . The biological neural system architecture is completely different from the Von neumann architecture . This difference significantly affects the type of functions each computational model can perform best which is shown in table 4.1.

Numerous efforts to develop " intelligent " program based on Von Neumann's centralized architecture have not resulted in any general – purpose intelligent programs inspired by biological neural networks . ANN are massively parallel computing systems consisting of an extremely large number of simple processors with many interconnections . ANN models attempt to use some " Organizational " principles believed to be used in the human brain

TABLE 4.1 VON NEUMANN COMPUTER VERSUS BIOLOGICAL NEURAL SYSTEM

	Von Newman Computer	Biological Neural System
Processor	Complex High speed One or a few	Simple Low speed A large number
Memory	Separate from a processor Localized Non content addressable	Integrated into processor Distributed content addressable
Computing	Centralized Sequential Stored programs	Distributed parallel self learning

Reliability expertise	Very vulnerable Numerical and symbolic Manipulations	Robust perceptual problems
Operating environment	Well – defined Well – constrained	Poorly defined unconstrained

Either human or other computer techniques can use neural networks , with their remarkable ability to derive meaning complicated to imprecise data, to extract patterns and detect trends that are to complex to be noticed . A trained neural network can be thought of as an expert in the category of information it has been given to analyze . This expert can then be used to provide projections given new situations of interest and answer what if questions.

Other advantages include :

Adaptive learning : An ability to learn how to tasks based on the data given for training to initial experience.

Self organization : An ANN can create its own organization or representation of the information it receives during learning time.

Real – time operation : ANN computations may be carried out in parallel , using special hardware devised designed and manufactured to take advantage of this capability .

Fault tolerance via redundant information coding : Partial destruction of a network leads to a corresponding degradation of performance . However some network capabilities may be retained even after major network damage due to this feature.

4.2 BIOLOGICAL NEURAL NETWORKS

A biological neuron or a nerve cell consists of synapses, dendrites , the cell body (or hillock) , and the axon . The building blocks are discusses as follows. The synapses are elementary signal processing devices.

A synapse is a bio chemical device , which converts a pre – synaptic electric signal in to a chemical signal and then back into a post – synaptic electrical signal.

The input pulse train has its amplitude modified by parameters stored in the synapse. The nature of this modification depends on the type of the synapse , which can be either inhibitory or excitatory.

The postsynaptic signals are aggregated and transferred along the dendrites to the nerve cell body. The cell body generates the output neuronal signal , a spike , which is transferred along the axon to the synaptic terminals of other neurons.

The frequency of firing of a neuron is proportional to the total synaptic activities and is controlled by the synaptic parameters (weights)

The pyramidal cell can receive 104 synaptic inputs and it can fan – out the out put signal to thousands of target cells – a connectivity difficult to achieve in the artificial neural networks.The terminologies of BNN and ANN is shown in table 4.2.

In general the function of the man elements can be given as,

Dendrite	-	Receives signals from other neurons
Soma	-	Sums all the incoming signals
Axon	-	When a particular of input is received then the cell frees .

It transmit signal through axon to other cells.

The fundamental processing element of neural network is a neuron . This building block of human awareness encompasses a few general capabilities . Basically , a biological neuron receives inputs from other sources , combines them in some way performs a generally nonlinear operation on the result , and then outputs the final result.Table 4.3 shows comparison between BNN and ANN.The properties of the biological neuron possess some features on the artificial neuron. They are,

1. Signals are received by the processing elements .This elements sums the weighted inputs .
2. The weight at the receiving end has the capability to modify the incoming signal.
3. The neuron fires (transmits , out put) when sufficient input is obtained.
4. The output produced from one neuron may be transmitted to other neurons.
5. The processing of information is found to be local .
6. The weights can be modified by experience.
7. Neurotransmitters for the synapse may be excitatory or inhibitory.
8. Both artificial and biological neurons have inbuilt fault tolerance.

TABLE 4.2 ASSOCIATED TERMINOLOGIES OF BIOLOGICAL AND ARTIFICIAL NEURAL NETWORK.

Biological Neural Network	Artificial Neural Network
Cell Body	Neurons
Dendrite	Weights or interconnections
Soma	Net input
Axon	Out put

TABLE 4.3 COMPARISON BETWEEN ARTIFICIAL AND BIOLOGICAL NEURAL NETWORK

Characteristics	Artificial neural Network	Biological neural network
Speed	Neural networks are faster in processing information . The cycle time corresponding to execution of one step of a program in the central	Biological neurons are slow in processing information. The cycle time corresponding to a neural event prompted by an external stimulus occurs in a milli second range.
Processing	Many program have large number of instructions , and they operate in a sequential mod one instruction after another on a conventional computer	Biological neural networks can perform massively parallel operations . The brain possesses the capability to operate with massively parallel operations , each of them having only few steps.

Size and complexity	These do not involves as much computational neurons . hence it is difficult to perform complex pattern reorganization	Neural networks have large number of computing elements, and the computing is not restricted to within neurons. The number of neurons in the brain is estimated to about 10^{11} and the total number of interconnections to be around 10^{15} The size and complexity of connections gives the brain the power of performing complex pattern reorganization tasks , Which cannot be realized on a computer.
Storage	In a computer , the information is stored in the memory ,which is addressed by its location . Nay new information in the same location destroys the old information. Hence her it is strictly replaceable	Neural networks store information in the strengths of the interconnections . Information in the brain is adaptable , because new information is added by adjusting the inter connection strengths , without destroying the old information.
Fault tolerance	Artificial nets are inherently not faulty tolerant, since the information corrupted in the memory cannot be retrieved	They exhibit fault tolerance since the information is distributed in the connections throughout the network . even though if few connections are not working the information is still preserved due to the distributed nature of the encoded information
Control mechanism	There is a control unit , which monitors all the activities of computing	There is no central for processing information in the brain . The neuron acts based on the information ;locally available , and transmits its output to the neurons connected to it. There is no specific control mechanism s external to the computing task.

4.5 BASIC BUILDING BLOCKS ARTIFICIAL NEURAL NETWORKS

The basic building blocks of artificial neural network are

Network architecture

Setting the weights

Activation functions

Network Architecture

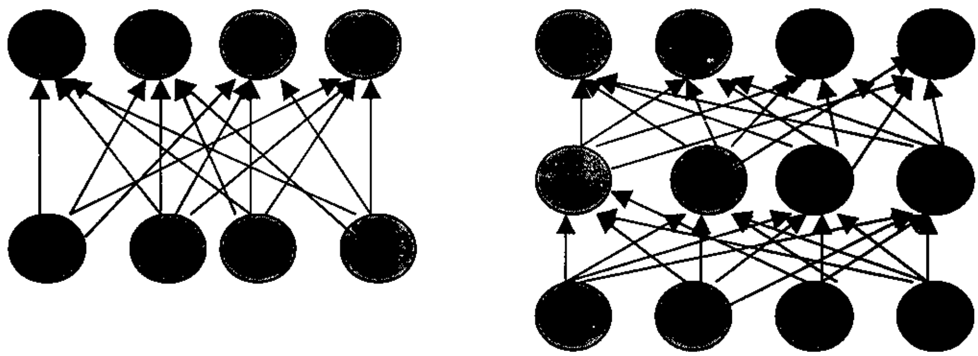
The arrangement of neurons into layers and the pattern of connection within and in between the layers are generally called as the architecture of the net . The neurons within a layer are found to be fully interconnected or not interconnected . The number of layers in the net can be defined to be the number of layers of weighted interconnected links between the particular slabs of neurons which is shown in figure 4.1. If two layers of interconnected weights are present , then it is found to have hidden layers. There are various type of network architectures : Feed forwards , feed back , fully interconnected net, Competitive net, etc.

Artificial neural networks come in many different shapes and sizes . In feed forward architectures , the activations of the input units are set and then propagated through the network until the values of the output units are determined . The network acts as a vector - valued function taking one vector on the input and returning another vector on the output .

For instance , the input vector might represent the characteristics of a bank customer and the output might be a prediction of whether that customer is likely to default on a loan or the inputs might represent the characteristics of a gang member and the output might be a prediction of the gang to which that person belongs.

A DISCUSSION ON SOME COMMONLY USED NETWORK

Output **Out put**



Input **Input**

Single layer feed forward **Multi layer feed forward**

FIGURE 4.1 TYPES OF NETWORK

Feed Forward Network

Feed forward networks may have a single layer of weights where the inputs are directly connected to the outputs , or multiple layers with intervening sets of hidden units . Neural networks use hidden units to create internal representations of the input patterns . In fact , the above figure 4.1 shows that given enough hidden units, its possible to approximate arbitrarily any function with a sample feed forwards network . This result has encourages people to use networks to solve many kind of problems .

Single Layer Network

It is a feed forward net. It has only layer of weighted interconnections. The inputs may be connected fully to the output units . But there is a chance that none of the input units and output units respectively . There is also a case where , the input units are connected with other input units and output units with other output units. In a single layer net, the weights from one output unit do not influence the weights for other output units.

Multi Layer Network

It is also a feed forward network i.e the net where the signals flow the input units to the output units in a forward direction . The Multi – layer network possess one or more layers of modes between the input and output units . This is advantages ever single layer net the sense that, it can be used to solve more complicated problems.

Setting the Weights

The method of setting the value for the weight enables the process of learning or training . The process of modifying the weights in the connections between network layers with the objective of achieving the expected output is called training a network . The internal process that takes place when a network trained is called learning . Generally there are three types of training as follows.

Supervised Training

Supervised training is the process of providing the network with a series of sample inputs and comparing the output with the expected responses . The training continues until the network is able to provide the expected response .In a neural net, for a sequence of training input vectors there may exist target output vectors . The weights may then be adjusted according to a learning algorithm .This process is called supervised training. In a logic circuit we might have the target output as +1, if the necessary logic condition is satisfied , or –1, if the logic condition is not satisfied . These type of logical nets are trained using supervises algorithm . The same criterion is applicable for pattern classification net also. Supervised training is adopted in pattern association as well if a neural net is trained to associate a set of input vectors with a corresponding set of output vectors with a corresponding set of output vectors , then it is called associative memory net.

If the output is same as the input , then it forms auto – associative memory , if the output is different from the input then it is hetero – associative.

Some of the supervised learning algorithms include heb net, pattern association memory net, back propagation net, counter propagation net, etc,

Unsupervised Training

In a neural net, if for the training input vectors , the target output is not known, the training method adopted is called as unsupervised training . The net may modify the weights so that the most similar input vector is assigned to the same output unit . The net is found to form a example or code book vector for each cluster formed .

Unsupervised networks are far more complex and difficult to implement . It involves looping connections back into feedback layers and iterating through the process until some sort of stable recall can be achieved . Un supervises networks are also called self-learning networks or self organizing networks because of their ability to carry out self – learning . This is the methods adopted in the case of self - organizing feature maps, adaptive resonance theory etc, the training process extracts the statistical properties of the training set and groups similar vectors into classes.

4.6 ARTIFICIAL NEURAL NETWORK (ANN) TERMINOLOGIES

The key terms used in the discussion on artificial neural networks are discussed below .

Weights

As discussed in the previous sections ,a neural network consists of a large number of simple processing elements called neurons . These neurons are connected to the each other by directed communication links, which are associated with weights .

" Weight is an information used by the neural net to solve a problem "

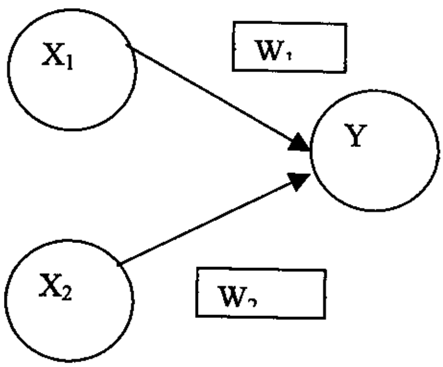


FIGURE 4.2 WEIGHT ALLOCATION TO HIDDEN LAYER

Figure 4.2 indicates a simple neural network . The weights that carry information are denoted by W_1 and W_2 . They may be fixed ,or can take random values, Weights can be set to zero or can be calculates by some methods . Initialization of weights is an important criteria in a neural net. The weight changes indicate the overall performance of the neural net.

X_1	=	Activation of neuron 1 (input signal)
X_2	=	Activation of neuron 2 (input signal)
Y	=	Out put neuron
W_1	=	Weight connecting neuron 1 to output
W_2	=	Weight connecting neuron 2 to output

Based on all these parameters , the net input ' Net' is calculated . The net is the summation of the products of the weights and the input signals.

$$\text{Net} = X_1 W_1 + X_2 W_2$$

Generally, it can be written as

$$\text{Net input} = \text{Net} = \sum X_i W_i$$

From the calculated net input , applying the activations functions , the output

4.7 ACTIVATION FUNCTION

The activation function is used to calculate the output response of a neuron . The sum of the weights input signal is applied with an activation to obtain the response. For neurons in same layer , same activation functions . The non linear activation functions are used in a multi layer net.

4.8 BIAS

A bias acts exactly as a weight on a connection from a unit whose activation is always 1 . Increasing the bias increases the net input to the unit ($b=w_0$)

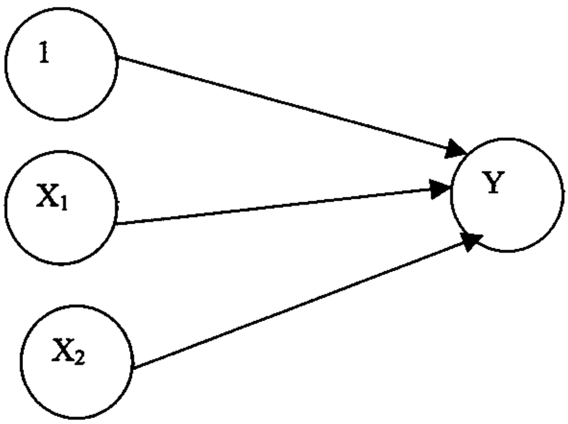


FIGURE 4.3 BIAS

The bias improves the performance of the neural network . Similar to initialization of weights , bias should also be initialized either to 0, or to any specified value , based on the neural net which is shown in figure 4.3. If bias is present then input is calculated as ,

$$Net = b + \sum X_i W_i$$

Where, net - Input

b - Bias

X_i - Input from neuron

Hence the activation function is obtained as ,

$$F(\text{net}) = \begin{cases} +1 ; \text{ if net} \geq 0 ; \\ -1 ; \text{ if net} < 0 ; \end{cases}$$

if bias is included

The threshold θ is a factor which is used in calculating the activations of the given net. Based on the value of threshold the out put may be calculated i.e the activation function is based on the value of θ for example the activation functions may be.

(i) $y = f(\text{net}) = +1$ if net $\geq \theta$
-1 if net $< \theta$

(ii) $y_i = f(\text{net}) = 1$ if net $y_{inj} > \theta$
-1 if net $y_{inj} < \theta$

hence θ and θ indicate the thresholds due to which the systems reponse is calculated . The threshold value is defined by the user.

4.9 Mc CULLOCH - PITTS NEURON MODEL

The first formal definition of a synthetic neuron model based on the highly simplified consideration of the biological model was formatted by Warren Mc – Culloch and Walter Pitts in 1943 . The Mc Culloch Pitts model of a neuron is characterized by this formalization , elegant and precise mathematical definition.

Mc Culloch – Pitts neuron allows binary 0 or 1 states only 1 states only i.e it is binary activated . These neurons are connected by direct weight path . The connected path can be excitatory or inhibitory . Excitatory connections have positive weights and inhibitory connections have negative weights . There will be same weights for the excitatory connection entering into a particular neuron.

The neuron is associated with the threshold value . The neuron fires if the net input if the neuron is greater than the threshold. The threshold is set so that the inhibition is absolute , because non zero inhibitory input will prevent the neuron from firing . It takes only one time step for a signal to pass over one connection link

Architecture

The architecture of the Mc - culloch pitts neuron is shown in Fig 4.4.

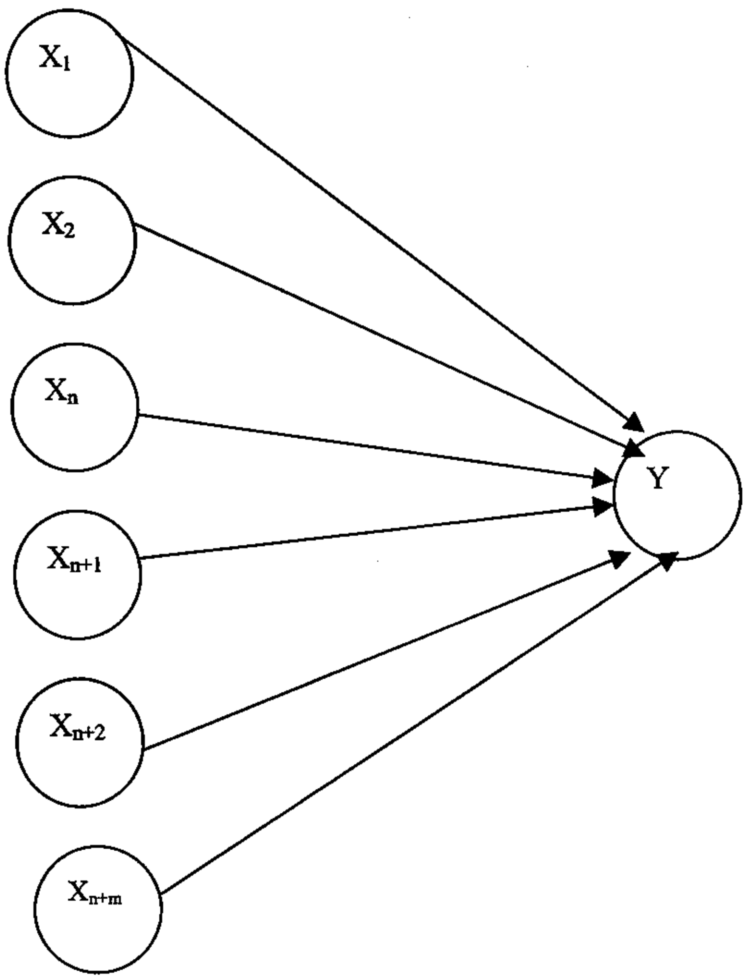


FIGURE 4. 4 A MODEL OF A NEURON.

Y is the Mc Culloch - Pitts neuron it can receive signal from any number of other neurons . The connections weights from $X_1 \ldots X_2$ are excitatory , denoted by 'W' and the connection weights from

X_{n+i} .... X_{n+m} are inhibitory denoted by '-p. The mc Culloch - pits neuron Y has the activation function .

$$F(y\,in) = \begin{cases} +1 & \text{if net} \geq \theta \\ -1 & \text{if net} < \theta \end{cases}$$

4.10 TRAINING ALGORITHM

Step 0: Initialize weight, (set of small random values).

Step1: While stopping condition are false, do steps 2-9.

Step2: For each training pair, do steps 3-8.

Feed Forward

Step 3: Each input unit (x_i, i=1, 2....p) receives input signal x_i and broadcast this signal to all units in the layer above (the hidden units)

Step 4: Each hidden unit (z_j, j=1, 2...p) sums its weighted input signal

$$Z_{inj} = Vo_j + \sum X_i V_{ij} ,$$

Applies its activation function to compute its output signal

$$Z_j = f(Z_{inj})$$

And sends this signal to all units in the layers above (output units)

Step 5: Each output unit (Y_k, k=1, 2, and 3....m) sums its weighted input signals.

$$Y_{ink} = W_{ok} + \sum Z_j W_{ij}$$

And applies its activation function to compute its output signal

$$Y_k = f(Y_{ink})$$

Back Propagation of Error

Step 6: Each output unit (Y_k , k=1,2,3,....m) receives a target pattern corresponding to the input training

Pattern, computes its error information term.

$$Ä_k = (t_k - y_k) f(Y_{ink})$$

Calculates its weights correction sum (used to update work later)

$W_{jk} = \alpha S_k Z_j$

Calculates its bias correction sum (used to update work later)

$W_{ok} = \alpha S_k$

And sends S_K units in the layer below.

Step 7: Each hidden unit (Z_j = j=1, 2…p) Sums its delta inputs (from units in the layer above)

$S_{inj} = \sum S_k W_{jk}$

Update weights and biases

Step 8: Each output unit (Y_k ,k =1, 2 .m) Update its bias and weights (j=0, 1, 2…p)

$W_{jk}(new) = W_{jk}(old) + W_{jk}$

Each hidden unit (Z_j ,j=1, 2….p) updates its bias and weights (I = 0,1,2….n)

$V_{ij}(new) = V_{ij}(old) + V_{ij}$

Step 9 : Test for stopping condition.

COMPUTATION

FEED FORWARD BACK PROPAGATION

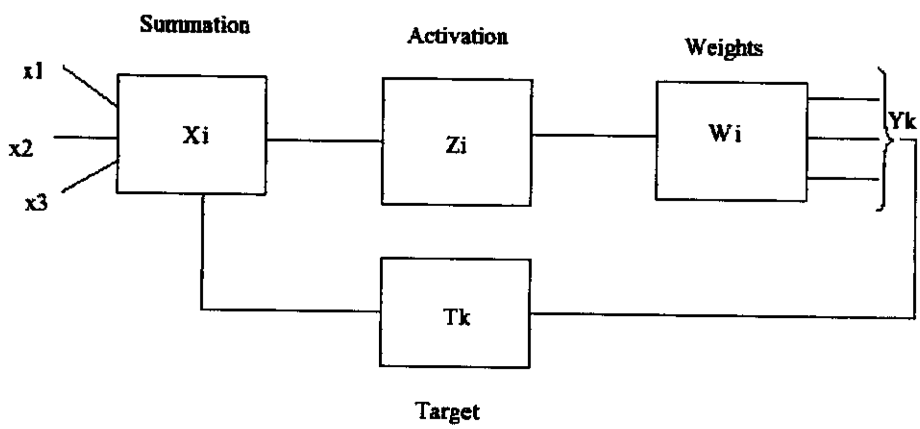


FIGURE 4.5 BLOCK DIAGRAM OF FFBP

Illustration: Adjustment of Weights of Connections From a Neuron to the Input Layer

Let us look at how adjustments are calculated for the weights on connections going from the j th neuron in the input layer to neurons in the hidden layer . Let us take specifically I=3, for illustration.

Much of the information we need is already obtained in the previous discussion for the second hidden layer neuron .

We have the errors in the computed output as the vector (- 0.09, - 0.16, 0.18,0.44) , and we obtained the error for the second neuron in the hidden layer as – 0.0041, which was not used above . just as the error in the output is propagated back to assign errors for the neurons in the hidden layer , those errors can be propagated to the input layer neurons .

The above figure 4.5 describes about the adjustment for the weights on connections between the input and hidden layers , we need the errors determined for the outputs of hidden layer neurons, a learning rate parameter , and the activation of the input neurons , which are just the input values for the input layer. Let us take the learning rate parameter to be 0.15. Then the weight adjustments for the connections from the third input neuron to the hidden layer neurons are obtain by multiplying the particular hidden layer neuron's output error by the learning rate parameter and by the input component from the input neuron . The adjustment for the weight on the connection from the third input neuron to the second hidden layer neuron is 0.15*3.2* -0.0041 which works out to –0.002.

If the weight on this connections is say - 0.45 , then adding the adjustment of - 0.002, we get the modified weight of –0.452, to be used in the next iteration of the network operation . Similar calculations are made to modify all other weights as well.

4.11 APPLICATIONS OF ANN

Neural networks have been successfully applied for the solution of a variety of problems however ,some of the common application domains have been listed below.

Pattern Recognition /Image Processing

Neural networks have shown remarkable progress in the recognition of visual images ,handwritten character, printed characters, Optimization/constraint satisfaction

This comprises problem which need to satisfy constraints and obtain solution. Examples of such problems include manufacturing scheduling, finding the shortest possible tour given a set of cities etc.

Several problems of this nature arising out of industrial and manufacturing fields have found acceptable solution using NNs

Fore Casting and Risk Assessment

Neutral networks have exhibited the capability to predict situations form the past trends. They have therefore , found sample applications in areas such as meteorology , stock market, banking , and econometrics with high Success rates.

Control Systems

Neutral networks have gained commercial ground by finding applications in control systems, dozens of computer products , especially by the Japanese speech and other PR based tasks companies incorporating NN technology is a standing example , Besides they have also been used for the control of chemical plants , robots and so , on.

4.12 FUZZY LOGIC

All the criteria characterizing a good solution can be taken into account by representing each criterion as a fuzzy set. The intuitive idea behind this is the greater the membership of the solution in the set ,the better the solution. The objective function is obtained from the aggregation of the fuzzy sets representing different criteria. Thus, the objective function includes every criterion affecting the solution .It is also possible to specify conflicting goals where different criteria draw the solution to different directions. The final solution is essentially a comprise among all the criteria. Also, the priorities among the criteria have to be weights ensure that the more important criteria have a greater effect on the value of the objective function than the less important ones.

In many cases a criterion can be considered as a fuzzy number. For example, the fuzzy set which corresponds to the criterion the group sizes should be as even as possible can be formulated as a fuzzy number N=m/n where 'm' is the total number of machine and 'n' the total number of jobs.

There are many ways to weight the criteria. One must bear in mind that the poorly fulfilled criteria affect the aggregated result more than the criteria with higher membership

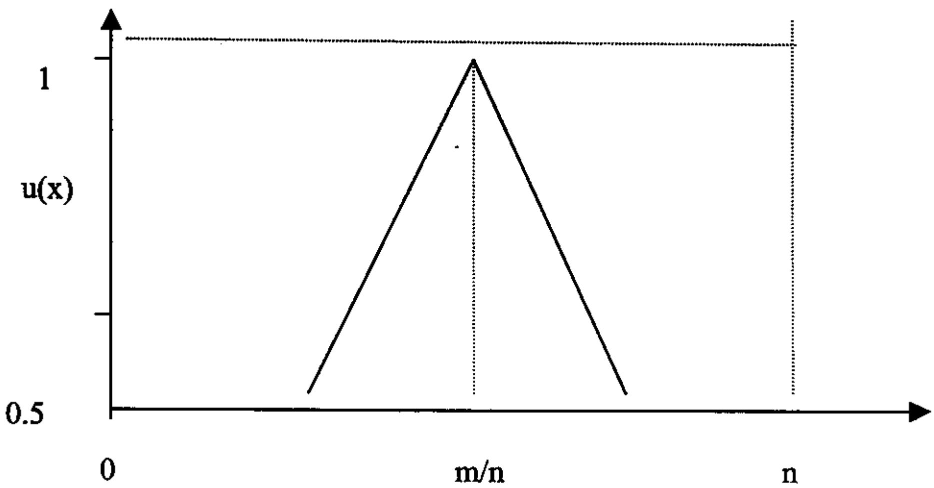


FIGURE 4.6 TRIANGULAR MEMBERSHIP FUNCTION

values is shown in figure 4.6. Therefore, weighing can be treated can be based on an interpretation of the fuzzy implication as a boundary which guarantees that a criterion has atleast a certain fulfillment value. Let us assume that a fuzzy criterion C_i has a weight W_i E[0,1] where a greater value corresponds to a higher priority. Thus, the weighted value of a criterion is obtained from the implication $W_i = C_i$

4.13 PREVIOUS APPROACHES TO FUZZY SCHEDULING

Fuzzy logic allows the modeling of imprecise scheduling knowledge with linguistic variables defined by membership functions showing the degree of presideness of the data and the reasoning about the imprecise data by using fuzzy rules. This is a means of representation and processing that is transparent and comprehensible for the user.

While the presented fuzzy – based approach is innovative within the field of scheduling there already exist several approaches to solve general or local scheduling problems by techniques founded on fuzzy set theory and fuzzy logic . Most of the approaches to solve scheduling problems use conventional (from operations research) or knowledge – based methods based on simple heuristics SPT, FCFS , LCFS , RANDOM , LPT that provide sufficient solution with respect to their evaluation functions . Wide used are the concepts of constraints and heuristics (in the form of rules) for both representation and solution of scheduling problems.

4.14 FUZZY SCHEDULING

To solve the scheduling problem with a fuzzy-based system the following sub - problems need to be addressed .

Modeling and transformation of scheduling data into a knowledge representation that can be handled by a fuzzy controller (fuzzification)

Processing of the fuzzy scheduling knowledge towards a decision by means of given rules and integration of fuzzy arithmetic to deal with imprecise or vague data and transformation of the fuzzy scheduling decision into crisp scheduling data (defuzzification).

SYSTEMATIC ARCHITECTURE OF FUZZY CONTROLLER

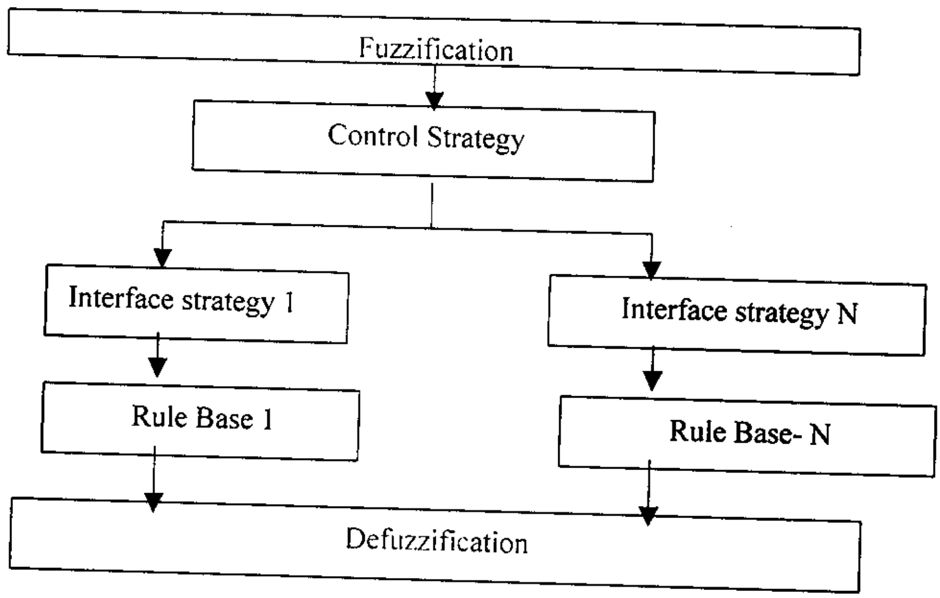


FIGURE 4.7 FUZZY INFERENCE SYSTEMS

The above figure 4.7 visualizes the schematic architecture of the fuzzy controller we use to solve the global predictive scheduling tasks . The data for the fuzzy based system is made available in an appropriate format by fuzzification . The rule bases together with the inference strategy determine the method for generating a predictive schedule.

4.15 MODELING OF FUZZY DATA IN THE SCHEDULING DOMAIN BY NEUTRALIZING FUZZY SYSTEMS

In a step the scheduling has to be classified and reduced to its main features to transform the scheduling data into a format that can be understood by a fuzzy based system . The notion of linguistic variables is used to describe the relevant features of the scheduling data qualifiedly. In principle , all data that are used in the model can be transformed into linguistic variables , but in many cases a discrete representation in sufficient.

Resources : no of machines, no .of jobs

Products : time consumption for a product

The following overview shows some of the modeled scheduling features and denotes the possible values of the linguistic variables . In most cases , triangular or trapezoidal membership functions are used ,because they can be implemented efficiently and the linear growth or decline of gradual membership regarding a value of maximum membership regarding a value of membership can be matched simply.

Processing times of jobs are uncertain due to both machine and human factors . Consequently, the completion time of each job is uncertain . In addition , as it not always possible to construct a schedule in which all jobs are completed before their due dates, some of the jobs may be tardy . The model should allow the decision make to express his / her preference to the jobs may be tardy . The model should allow the decision make to express his / her preference to the tardiness of each job . Fuzzy set are used to model uncertain processing times of jobs and the decision maker's preference top the tardiness of each job.

Unlike a conventional crisp set, which enforces either membership or non membership of an object in a set , a fuzzy set allows grades of membership of an object in a set , a fuzzy set allows grades membership in the set. A fuzzy set A is defined by a membership function which assigns to each object x in the universe of discourse X , a value representing its grade of membership in this fuzzy set A variety of shapes can be used for membership such as triangular , trapezoidal , bell curves and s-curves , conventionally , the choice of the shape is subjective and allows the decision maker to express to express his /her preferences. The estimation of processing time of each operation is obtained taking into consideration the nature of the machines in use. While some machines are automated and can be operated at different speeds, others are staff – operated and therefore the processing times are staff- dependent .

We mean to introduce neural concepts into fuzzy systems. Technically, it any be realized by mapping a fuzzy system into a neural network , either

either disturbed or localized . In the former case , the input - output equivalence of both systems is of primary concern.

Knowledge traditionally represented by IF – THEN rules is implicitly and distributive represented by connection weights and local processing units in the network . Each unit is responsible for many rules with different activation strengths . In other words, the function of each IF – THEN rule is undertaken by many units. It seems that a disturbed network inherently possesses some fuzziness which may come from the distributiveness of the network although the term fuzziness here is used more technically and less conceptually . On the other hand , instead of paying primary attention to the functional aspect, the localized method is concerned mainly with the structural aspect , that is , to seek a close structural equivalence between the logic - based algorithm and the neural network structure . In this regard , each IF – THEN rule can be represented by only one unit in the network with associated weight vectors accounting for the IF part and THEN part . Thus knowledge is represented ever, the computing (reasoning) process for deriving an appropriate action in response to an input may be competitive or co –operative depending s on the corresponding reasoning algorithms .

CHAPTER 5

WORKLOAD ALLOCATION USING ANN

The parallel machine scheduling problem consists of a set of 'm' machines M_j ,j=1..m and a set ' n' jobs . i = 1 ..n ,J_i requires processing time P_j can be processed on any one of the machines

The processing times of all the jobs are multiplied with weight using NN the input layer and then sent to the hidden layer which is setting as machine's . The output of the hidden layer is noted for various strategies. After the jobs get allocated to the machine the RPI is calculated for all the strategies . The threshold value is set and is stored in a separate value after comparing the results the least strategy is find out using NN logic by C programming.

When a machine becomes available a job form the list of unscheduled jobs with the highest priority based in RANDOM , SPT and LPT, FIFO , LIFO is selected and assigned to the machine for processing . Many simple and mixed priority rules were developed and tested in a number of research papers to meet the objectives of scheduling . Many varied objectives have also been studied for parallel machine scheduling.

This research focuses on evaluating the performance of scheduling strategies on parallel machine scheduling with the measure of workflow balancing among the machines . Instead of assigning a job to the first available machine M_k the machine with less cumulative workflow is selected for a job selected from the unscheduled list of jobs . The workflow W_k of machine M_k is calculated by summing all the processing times of jobs . J_i assigned to it. Let S_i be the sets of scheduled jobs to the machines M_i , i – 1 , m respectively . hence the work flow of any machine can be found using

$$Wi = \sum_{J=i}^{n} pj \qquad \text{if } j = S_i \ (I=1\ldots m)$$

Where, S_i - set of scheduled jobs.

5.1 WORK LOAD ALLOCATION ALGORITHM

The work load allocation algorithm using NN is classified below

Step 1: The input data from NN is assigned to the machines based on the strategies

Step 2: The weight limitations is considered using fuzzy output.

Step 3: Select the following strategies one by one & repeat steps.

Strategy 1 (Shortest Processing Time – SPT)

Arrange the list of unscheduled jbs of set A in increasing order of processing times P_i

Strategy 2 (Longest Processing Time - LPT)

Arrange the list of unscheduled jobs of set A is decreasing order of processing times P_i

Strategy 3 (Random processing Time – RANDOM)

Use the list of unscheduled jobs of set A to select random processing time

Strategy 4 (First In First Out – FIFO)

Arrange the list of unscheduled jobs of set A that entered first in the order of processing times.

Strategy 5 (Last In First Out - LIFO)

Arrange the list of un scheduled jobs of set A that entered last in the order of processing times

Step 4: Assign the jobs to machine based on step1

Step 5: Find out the cumulative of each machines i.e sum of all processing times of the jobs assigned to the particular machine.

Step 6: Select a machine with less cumulative workload among Machines available.

Step 7: Find out the mean flow time of each machine from the set of jobs i.e sum of all processing times of the jobs assigned to the particular set of m machines.

Step 8 : Find out the makes pan from the set of jobs that is assigned to the machines.

Step 9: Repeat steps 2 to 6 until all the jobs from the list A are assigned .

CHAPTER 6

SIMULATION MODEL

This shows the sequence of jobs assigned to a machine in terms of job processing times. The sequence of jobs can be extracted from array B. Array C is used to store the cumulative work load for all the jobs assigned to the machines. Here rows indicate the machines and columns represents the cumulative workload of the machines. The cumulative workload of a machine is equal to the sum of all processing times of jobs assigned to that machine. The cumulative workload of all machines is compared and a machine with lesser workload is selected for assigning the next job from the ordered list of jobs stored in array D. This work assignment process continues until all the jobs from the list are assigned. This mode takes number of jobs and machines as input.

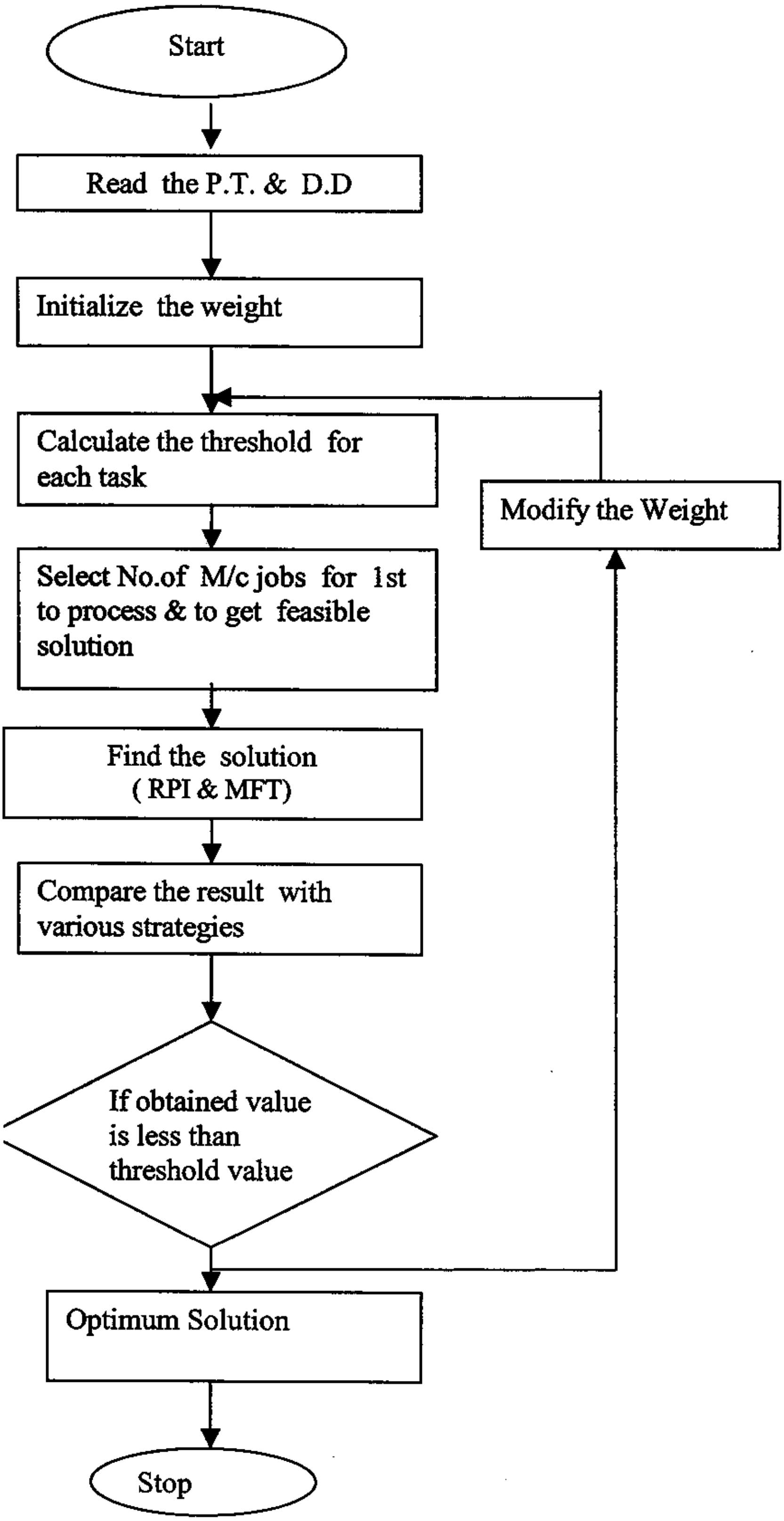
The processing times of jobs stored in array A were used for the RANDOM strategy and were transferred to array D. Using a bubble-sorting algorithm, previously stored processing times in array A were ordered in increasing order of job processing times for SPT and decreasing order of job processing times for LPT strategies and then stored in array D. Then the processing times of all the jobs assigned to a particular machine or row were added together and stored in array C in their respective rows. The workload of each machine was compared to find out the row that represents a machine with a smaller added processing time ,i.e. workload. Jobs were then selected one by one from array D and assigned to the rows of array B that represent the machine with the lesser cumulative workload. This loop continues until all the jobs are assigned from array D to array B.

The simulation model based on neural network technique in the C – language on the IBM / PC compatible system for simulating test problems used for the study. The built in random number generator of the compiler is used for generating a sufficient number of instances .

The weight are generated between (0,1) to added with the processing times of jobs. The weighed processing times for jobs are stored in array E was used for allocating the jobs to all the machines according to various scheduling strategies .

The generated processing times are stored is an array A with job number can be address according to the required scheduling strategies in array K, 1^{st} column represent the machine and 2^{nd} column represent the processing times of jobs assigned to the machines .

The jobs processing times are then arranged in the order of FCFS & LCFS basic and the stored in array D. The below figure 6.1 visualizes the flow chart for Neural network, which determines the allocation of the jobs to machines based on five strategies.


Start
Read the P.T. & D.D
Initialize the weight
Calculate the threshold for each task
Modify the Weight
Select No.of M/c jobs for 1st to process & to get feasible solution
Find the solution (RPI & MFT)
Compare the result with various strategies
If obtained value is less than threshold value
Optimum Solution
Stop

6.1 PERFORMANCE MEASURES

Scheduling strategies were used for the selection of jobs for allocation on machines in parallel machine scheduling. Shop floor performance is mostly affected by these scheduling strategies. In order to achieve higher productivity on the shop floor, an appropriate strategy needs to be followed based on al the machines are calculated and compared with the maximum workload of a particular machine. The machine with maximum workload is the bottleneck, which affects the shop floor performance. In this study, the maximum load W_{max} on a machine is taken as the index for comparing the work load on other machines for evaluation purposes. The difference between the maximum workload and the available workload on a machine is called an imbalance in workload. Imbalances in machine workloads are expressed as

$$I_i = W_{max} - W_i \ , \ (i = 1 \ldots m)$$

The various workload balancing strategies are adopted to minimize these imbalance on the machines. The best schedule is the uniform distribution on the machines. The best schedule is the uniform distribution of workloads on all the machines. A careful study of the allocation of work in the bottleneck machine contributes towards the improvement of shop floor performance.

The allocation of jobs may be altered on the machines to minimize the maximum workload on the bottleneck machines . Some of the allocated jobs from the bottleneck machine can be moved to other imbalances machines to minimize the work load on it .

The relative percentage of imbalances (RPI) in workload of all the machines is adopted to compare the performance of the scheduling strategies .

This indicates the percentage of deviation of workloads of machines from the upper bound of maximum workload. It is expressed as

Relative percentage of imbalance (RPI) or

Percentage of deviation from upper bound $= \dfrac{(W\,max - Wi)}{W\,max} \times 100$

The average RPI values are calculated using Excel using the out put of the simulation model . The output of the simulation model shows the work allotments to the machines in terms of job processing times .The five scheduling strategies produce five different work allotment schedules .The maximum workload on the W_{max} machine is identified and the percentage of deviation form this maximum workload for all the machines is calculated individually for each of the five rules. The workload is balanced perfectly if the percentage of imbalance among the machines is zero.

The objectives of this paper is to minimize the maximum of imbalances existing with parallel machine scheduling . The best allocation of jobs amongst the machines can be obtained when the workloads are balanced . The performance measure of relative percentage of imbalance in workload (RPI) to evaluating the performance of the above discussed Scheduling strategies RANDOM , SPT,LPT, FCFS and LCFS in parallel machine scheduling.

6.2 COMPUTATIONAL RESULTS

The maximum workloads on machine for each strategy were then found . These maximum workloads were compared with the workloads of each machine .The difference between the maximum and available workloads on each machine was then calculated . This measure indicates the imbalance of work loads on the machines .Considering the maximum workloads as an agent , the relative percentage of imbalance for each machine was calculated for five different strategies were calculated for a set of n jobs.

The experiment was conducted on IBM / PC compatible system 3.5 data's have been used for a set of 'n' jobs and 'm' machines .

The number of jobs in each set was varied & the total number of sets considered for the study was seven . Then the total amount of instances generated were equal to 35 X 5 = 245 for m- machines problem .

Thus, 5 sets of m machine problems were considered for the computational experiment . Hence 245 x 5 = 1225 instances were generated to test the performance of the scheduling policies . The output of all these instances were cliental to excel worksheet to determine the graphical form.

40 jobs are taken as input which is shown in table 6.1 for NN and 10 jobs are taken as input for Fuzzy Logic which is shown in table 6.2.

TABLE 6.1 INPUT DATA FOR NEURAL NETWORK

Jobs	Processing Time	Due date
1	2	6
2	3	4
3	4	6
4	6	10
5	8	11
6	3	12
7	2	4
8	7	26
9	3	18
10	6	16
11	2	25
12	8	30
13	5	4
14	3	2
15	6	1
16	4	6
17	9	12
18	3	4
19	3	14

20	6	18
21	6	20
22	8	22
23	9	17
24	9	13
25	9	24
26	10	32
27	11	21
28	11	16
29	13	9
30	13	11
31	13	3
32	13	8
33	14	23
34	15	18
35	16	1
36	16	21
37	17	20
38	11	10
39	10	6
40	11	16

TABLE 6.2 INPUT DATA FOR FUZZY LOGIC

Jobs	Processing Time	Date dates
1	32	216
2	75	53
3	80	681
4	80	288
5	93	66
6	107	33
7	109	445
8	115	716
9	136	424
10	144	424

2 Jobs * 2 Machines

A parallel machine scheduling problem is considered for 4 Jobs with 2 Machines using Neural Network. The average Relative Percentage of Imbalance (RPI) of workloads for various strategies using 2 machines is shown in table 6.2.3. The maximum of RPI values of 2% is shown by LPT for Job size of 4. The other four rules are exhibiting lesser RPI values for the same Job size. Thus the Random and SPT rule approaches nearer to zero values. This indicates a perfect balance of workloads on all the four machines. The LPT rule produces higher RPI and Mean Flow Time of 0.5% and 2%.

Random rule produces less RPI values of below 1% for n=4.Thus SPT strategy produces good schedule with lesser RPI values of below 1% for 2 machine problem with Job size of 4.

TABLE 6.3 NEURAL RESULTS FOR 4 JOBS

LPT

S.NO	Pi	Wpi	TCT	MFT	RPI
1	2	1.8			
2	3	1.2			
3	4	0.6			
4	6	0.4	2	0.5	2.47

SPT

S.NO	Pi	Wpi	TCT	MFT	RPI
1	2	0.6			
2	3	0.9			
3	4	1.2			
4	6	1.8	2	0.5	0.669

LCFS

S.NO	Pi	Wpi	TCT	MFT	RPI
1	2	1.8			

3	4	0.9			
4	6	0.6	1.5	3.75	1.2

RANDOM

S.NO	Pi	Wpi	TCT	MFT	RPI
1	2	1.8			
2	3	1.2			
3	4	1			
4	6	0.4	2	0.5	0.95

FCFS

S.NO	Pi	Wpi	TCT	MFT	RPI
1	2	1.8			
2	3	1			
3	4	0.6			
4	6	1.4	2	0.5	1.47

RESULT

Best Strategy : SPT
Mean Flow Time : 0.5
RPI : 0.669

6 Jobs * 3 Machines

The behavior of five scheduling strategies on three machines are illustrates in table 6.2.4. The average RPI values for all the five strategies are shown in table. The RPI values of all the five strategies produces less than 0%.This indicates an perfect schedule when the Job size increases. The SPT rule starts with RPI values of 1% for n=4 and move below zero for the job size of n=6. When the job size increases, the RPI values decreases for all five rule. The values of five strategies will never coincides. They behave independently without crossing one another. There is a wide gap between SPT and LPT rule. But, SPT and FCFS are nearer to each other for job size of n=6. The SPT rule can be adopted for reducing the workload for scheduling the jobs on three parallel machines.

TABLE 6.4 NEURAL RESULTS FOR 6 JOBS

LPT

S.No	Pi	Wi	TCT	MFT	RPI
1	5	1.4			
2	6	1.2			
3	7	1			
4	8	1.6			
5	9	1.8			
6	10	1.3	9	1.5	0.0064

SPT

S.No	Pi	Wi	TCT	MFT	RPI
1	5	0.8			
2	6	0.6			
[illegible]	[illegible]	[illegible]			

4	8	1			
5	9	0.8			
6	10	1.2	5	1	0.0002

LCFS

S.No	Pi	Wi	TCT	MFT	RPI
1	5	3			
2	6	2.7			
3	7	2.4			
4	8	2.1			
5	9	1.8			
6	10	1.5	45	5.8	0.0039

RANDOM

S.No	Pi	Wi	TCT	MFT	RPI
1	5	3.1			
2	6	1.8			
3	7	1.5			
4	8	3.0			
5	9	2.7			
6	10	2.4	10	2	0.0004

FCFS

S.No	Pi	Wi	TCT	MFT	RPI
1	5	3.0			
2	6	2.7			

4	8	2.10			
5	9	1.8			
6	10	1.5	10	2	0.0003

RESULT

Best Strategy : SPT

Mean Flow Time : 1

RPI : 0.0002

8 Jobs * 4 Machines

Table 6.2.5 shows variations of workload on four machines against five scheduling strategies. All the five strategies starts with the RPI value of between 7% to 3% for n=8. The RPI values of LCFS, RANDOM and FCFS are approximately equal as the job size increases. Still, SPT rule produce perfect result with the RPI value of 0.0001 and LPT produces higher schedule of 0.0007.

TABLE 6.5 NEURAL RESULTS FOR 8 JOBS

LPT

S.No	Pi	Wi	TCT	MFT	RPI
1	2	2.7			
2	8	2.4			
3	5	1.8			
4	3	1.5			
5	6	1.2			
6	4	0.9			
7	9	0.9			
8	3	0.6	6	0.75	0.0007

SPT

S.No	Pi	Wi	TCT	MFT	RPI
1	2	0.6			
2	8	0.9			
3	5	0.9			
4	3	1.2			
5	6	1.5			
6	4	1.8			
7	9	2.4			
8	3	2.7	2	0.25	0.0001

LCFS

S.No	Pi	Wi	TCT	MFT	RPI
1	2	2.7			
2	8	2.4			
3	5	1.8			
4	3	1.5			
5	6	1.2			
6	4	0.9			
7	9	0.9			
8	3	0.3	40	5	0.00037

RANDOM

S.No	Pi	Wi	TCT	MFT	RPI
1	2	1.2			
2	8	0.9			
3	5	0.9			
4	3	0.6			
5	6	2.7			
6	4	2.4			
7	9	1.8			
8	3	1.5	6	0.75	0.00035

FCFS

S.No	Pi	Wi	TCT	MFT	RPI
1	2	2.7			
2	8	2.4			
3	5	1.8			
4	3	1.5			
5	6	1.2			
6	4	0.9			
7	9	0.9			
8	3	0.6	6	0.75	0.000357

RESULT

Best Strategy : SPT

Mean Flow Time : 0.25
RPI : 0.0001

10 Jobs * 5 Machines

Table 6.2.6 shows the results for the simulated problems on five parallel machines. The RPI value for all the five strategies for job size of n=10 decreases when compared to other job sizes. For all the job sizes, the RANDOM rule shows a moderate result and SPT rule produces perfect schedule.

TABLE 6.6 NEURAL RESULTS FOR 10 JOBS

LPT

S.No	Pi	Wi	TCT	MFT	RPI
1	3	3.9			
2	6	3.9			
3	6	3.3			
4	8	3.6			
5	9	2.7			
6	9	2.7			
7	10	2.4			
8	11	1.8			
9	13	1.8			
10	13	1.8	15	1.5	0.00098

SPT

S.No	Pi	Wi	TCT	MFT	RPI
1	3	1.8			
2	6	1.			
3	6	2.4			
4	8	2.7			
5	9	2.7			
6	9	3.0			
7	10	3.3			
8	11	3.9			
9	13	3.9			
10	13	2.9	4	0.4	0.0004

LCFS

S.No	Pi	Wi	TCT	MFT	RPI
1	3	3.9			
2	6	3.9			
3	6	3.3			
4	8	3.0			
5	9	2.6			
6	9	2.4			
7	10	2.3			
8	11	1.8			
9	13	1.7			

RANDOM

S.No	Pi	Wi	TCT	MFT	RPI
1	3	2.7			
2	6	2.4			
3	6	1.8			
4	8	1.6			
5	9	1.8			
6	9	2.8			
7	10	3.9			
8	11	3.9			
9	13	2.0			
10	13	1.6	15	1.5	0.00265

FCFS

S.No	Pi	Wi	TCT	MFT	RPI
1	3	2.7			
2	6	2.4			
3	6	1.8			
4	8	1.8			
5	9	1.8			
6	9	2			
7	10	3.9			
8	11	3.8			
9	13	2.0			

FCFS

S.No	Pi	Wi	TCT	MFT	RPI
1	3	2.7			
2	6	2.4			
3	6	1.8			
4	8	1.8			
5	9	1.8			
6	9	3.9			
7	10	3.4			
8	11	3.6			
9	13	2.0			
10	13	1.6	15	1.5	0.00275

RESULT

Best Strategy : SPT

Mean Flow Time : 0.4
RPI : 0.0004

CHAPTER 7

RESULTS AND DISCUSSION

The RPI values, make span & mean value flow time of variable job sizes using Neuro - fuzzy approach are compared with HM- Algorithm . The result shows that Neuro - Fuzzy approach gives the better results , when compared with the HM algorithm. The performance of various strategies using Neural network is discussed below. While comparing the strategies , the SPT rule shows lesser percentage of imbalance and Mean Flow Time for all the m Machine problems. The SPT rule gives least values of RPI below 1 % for job sizes of n = 4 in 2 machines, n=6 in 3 machines , n = 8 in four machines , n=10 in five machines problems . This indicates that when the number of machines for scheduling in increased , the lowest RPI values fall below 1 % for the increased job sizes . A shift of this RPI values on the right hand side is observed with an increase in n . All the five strategies start with higher RPI values for smaller range of n . The RPI values decreases for medium range of n & a higher of n . The SPT rule seems to produce good schedule with a lesser percentage of imbalances for all the m machines problems . After reaching the minimum of less than 1 % in all five cases it forms a straight line & involves horizontally along the x- axis.

These RPI values of SPT are less than the RPI values produced by other four rules. It is evident that the LCFS rule does not produce a good schedule with respect to workloads when the jobs size is lower on m machines .The RANDOM line lie between LCFS & LPT, lines LCFS & LPT lies between SPT & FCFS in all the cases , all the five rules shows a higher percentage of imbalance in work loads on machines with a smaller n . The research indicates that SPT strategy produces balanced workloads in parallel machine scheduling irrespective of job size & no of machines.

n Jobs x m- Machines

A parallel Machining scheduling problem is considered for various job sizes of n with m machines. The average relative percentage of imbalance (RPI) of workloads in different machines are shown in table 7.1 . A computational experiment was conducted for various job sizes of 5 to 40 on different machines . Five lines are plotted separately for each strategy viz RANDOM , SPT , LPT, FCFS and LCFS . The lines exhibit similar behaviour for the five different strategies . From fig 7 one can observe that all five different strategies produce similar trends with the increase in job size but with different RPI values. Workload deviations on machines are clearly shown by the measure of average relative percentage of imbalance on Y – axis .

The maximum of RPI value is shown for FCFS for the variable job sizes . The other four rules are exhibiting lesser RPI values for the different job size. The average RPI values decrease with increase in job sizes all the five strategies.

The lines show a higher percentage of imbalance for different job sizes. The SPT rule always exhibits lesser RPI values when it is compared with the other rule. The LCFS line approaches below few for the job sizes of more than 40. The LPT & FCFS rule produces higher RPI values on one end and LCFS & RANDOM gives lesser RPI values on the other end. The RANDOM rule produces good schedules with moderate RPI values. The SPT strategy produces good schedules with lesser RPI values for m machines and n job sizes.

TABLE 7.1 RPI RESULTS FOR VARIOUS STRATEGIES

Jobs	Random	LCFS	FCFS	LPT	SPT
4	20.64	19.34	17.32	13.13	10.16
6	18.11	17.94	15.12	12.13	9.54
8	16.00	15	13.89	11.68	8.76
10	15.43	14.87	11	8.45	5.4
15	13	11.9	8.76	6.93	4.2
20	11.3	9.8	7.2	5.4	3.2
25	9.8	5.3	4.19	4	2.9
30	7	3.7	3.5	3.6	9
35	6.4	3.	3.5	3.0	1.45
40	2.47	1.2	0.95	0.67	0.21

A Comparison of Neuro-fuzzy results with Hm – Algorithm

NN – Fuzzy

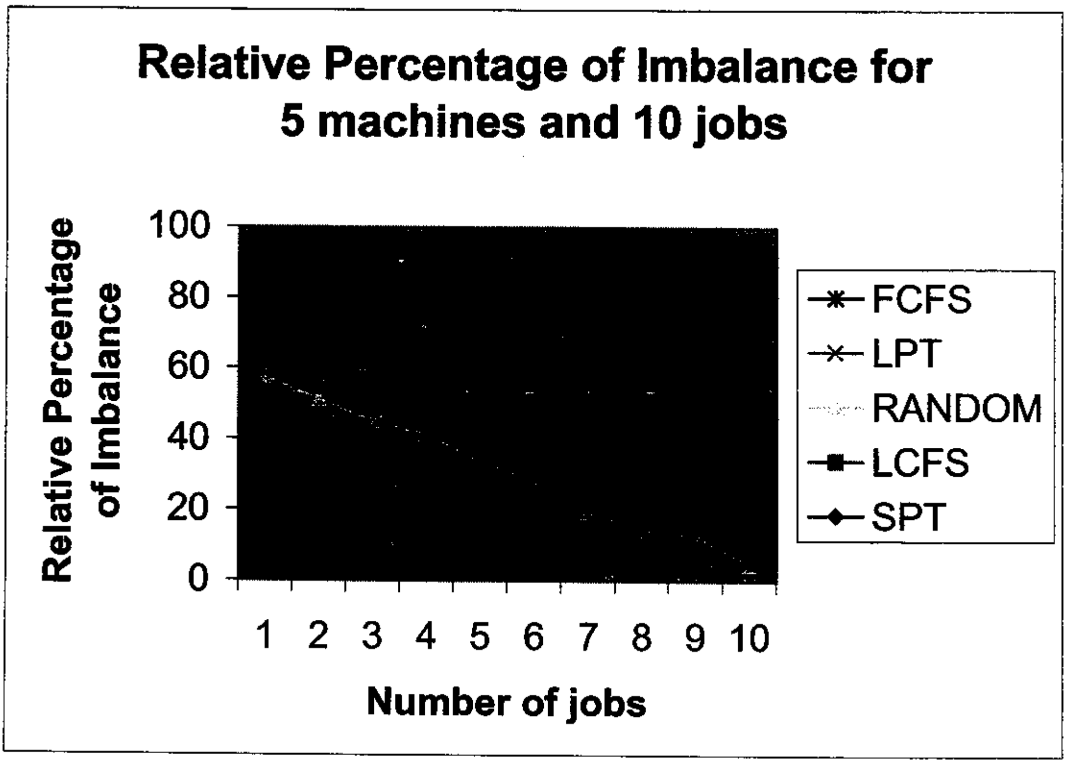
Mean Flow time = 0.25
RPI = 0.0001

Hm – Algorithm

Mean Flow time = 13
RPI = 4.823

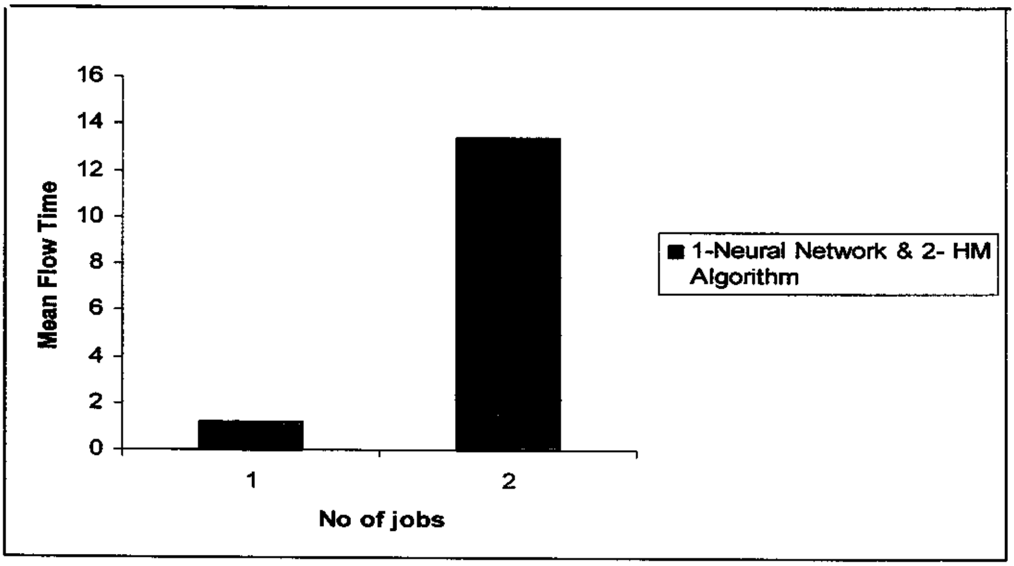
Figure 7.1 shows the results for the simulated problems on five parallel machines for job size of 40. The FCFS rule starts with a maximum RPI of 8% and linearly decreases i.e nearer to zero value. The LCFS starts with an average RPI of about 4% and reaches zero value to make a good schedule. Comparing to all the five strategies SPT starts with a perfect schedule with least imbalance of about 2% and decreases linearly to zero for job size of 40 and with machine

FIGURE 7.1 RPI FOR 5 MACHINES AND 10 JOBS



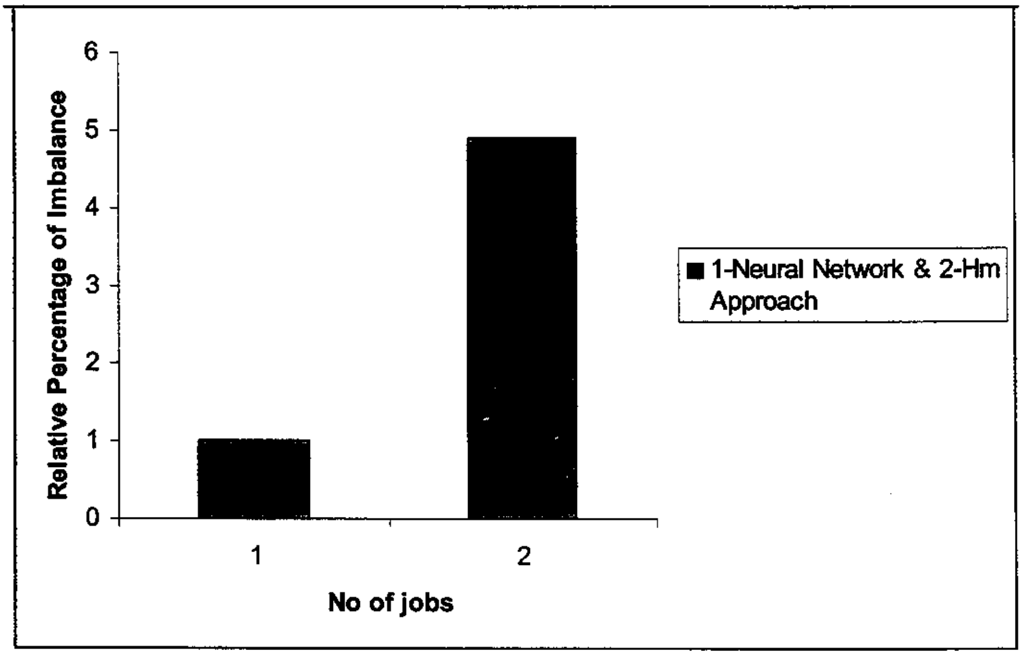
In figure 7.2, the Mean Flow Time of Neuro-Fuzzy approach is compared with Heuristic, and the results shows that Neuro-Fuzzy approach takes least time to give better performance for the jobs when compared with Heuristics.

FIGURE 7.2 NEURAL NETWORK VS HM ALGORITHM FOR MEAN FLOW TIME



In figure 7.3, the RPI values of Neuro-Fuzzy is compared with Heuristic and the result shows that Neuro-Fuzzy gives minimum RPI, to balance the workload on the on the machines when compared to Heuristics.

FIGURE 7.3 NEURAL NETWORK VS HM ALGORITHM FOR RELATIVE PERCENTAGE OF IMBALANCE



CHAPTER 8

CONCLUSION

In this project work an effort has been made by proposing a neural network framework for solving the parallel machine scheduling problem. The problem considered is that of optimizing the makespan, reducing the bottleneck thereby minimizing the RPI for scheduling n tasks on m identical machines. By varying the weights of makespan, the optimal performance measures are found, and the best-weighted combination is identified. The weights that are uncertain in NN are being trained using Fuzzy logic approach. These optimal solutions are used for training the neural network. Through weights assigned to the links between tasks and machines, and by adjusting the weights using an appropriate learning strategy, a significantly improved schedule is found using ANN.The data has been trained using c-program with NN logic. Further, a validation of the proposed methodology has been done using HM-Algorithm and the results with minimum Mean Flow Time and minimum Relative percentage of Imbalance have been got for Shortest Processing Time (SPT). It has been observed that the Neuro-fuzzy models perform better when compared to heuristic methods.

REFERENCES

1. Jatinder N.D.Gupta (2000 a), "Selecting Scheduling Heuristics Using Neural Network", *INFORMS Journal on Computing,* 12, pp. 2 –6.

2 Anurag Agarwal, Hasan Pirkul, Varghese S.Jacob (2002). "Augmented Neural Networks for Task Scheduling", *European Journal of Operational Research,* 15,pp.11-14.

. Harvey B.Newman, Josif C.Legrand (2000), "A Self-Organizing Neural Network for Job Scheduling in Distributed Systems",*International Journal of Advanced Manufacturing Technology,* 10, pp. 9-14.

Bruno, J. L., E. G. Coffman, and R. Sethi, (1974) "Algorithms for Minimizing Mean Flow Time," *IFIPS Congress* , 74, pp. 504-510.

Coffman, E. G. and R. Sethi , (1976). "Algorithms Minimizing Mean Flow Time: Schedule Length Properties," *Acta Informatica,* **6**, pp.1-14

Coffman, E. G., M. R. Garey, and D. S. Johnson, (1978). "An application of Bib-Packing to Multi- Processor Scheduling," *SIAM Journal on Computing,* **7**, pp.1-17

Eck, B. T. and M. Pinedo, (1993). "On the Minimization of the Makespan Subject to Flow Time Optimality," *Operations Research,* 41, pp .797-801

Gupta, J. N. D. and J. C. Ho, (2001). "Two Machines Hierarchical Scheduling with Makespan and Flow Time Criteria," *Computers and Operations Research*, 28, pp . 705-717

. Gupta, J. N. D. and A. J. Ruiz-Torres, (2000 b) "Minimizing Makespan Subject to Total Flow time on Identical Parallel Machines," *European Journal of Operational Research,* 125, pp. 370-380.

0. Leung, J. Y-T. and G. H. Young, (1989) "Minimizing Schedule Length Subject to Minimum Flow time," *SIAM Journal on Computing,* 18, pp. 314-326.

1. Saravanan.R, Asokan.P and Vijayakumar.K (2003) " Machining Parameters Optimization for Training Cylindrical Stock into a Continuous Finished Profile Using Genetic Algorithm and Simulated Annealing" , *International Journal of Advanced Manufacturing Technology*, 21, pp. 1-9.