International Journal of Engineering and Technical Research (IJETR)
ISSN: 2321-0869, Volume-3, Issue-4, April 2015

Earliness and Tardiness Single Machine Scheduling Problems Using Neural Network

Abdelaziz Hamad, Mohamed Hassen Elzubair

***Abstract*—In this paper we consider earliness and tardiness single machine scheduling problems with common and distinct due date problem based on artificial neural network. Scheduling problems involving both earliness and tardiness costs become more important in recent years. This kind of problems is classified as NP-hard problem. The objective is to sequence the jobs on the machine so that the total costs of earliness and tardiness be minimized. We developed a new model of multi layer back propagation neural network to find an optimum or near optimum solution for the problems. In our study neural network has been proven to be effective and robust in generating optimal or near optimal solutions to the problems Further research can be carried out for multi machines one.**

***Index Terms*— Single machine; Earliness Tardiness; Multi layer neural network**

I. Introduction

Job scheduling has been long known problem that is highly nonlinear and often formulated as a discrete combinatorial optimization problem with large interacting degrees of freedom. It is further complicated by a set of rules in the form of constraints that need to be satisfied which make the problem more difficult to solve. In general, for the job scheduling problem, which is classified as NP-complete in nature (Aho *et al.*, 1974) polynomial solutions are impossible in most cases due to large interacting degrees of freedom. Several methods, such as enumerated search and mathematical programming, have been proposed to solve the problem. Polynomial solutions have been obtained in some restricted cases. Many heuristics have been proposed in recent years, such as list scheduling heuristic, tabu search, simulated annealing, genetic algorithms and artificial neural network.

In particular the study of the earliness and tardiness scheduling problems are a relatively recent areas of inquiry. A job completed before the due date called is an early job and after the due date is called a tardy job. For many years, scheduling research focused on single performance measures.. Most of the literature deals with such regular measures as mean flow time, mean lateness, percentage of jobs tardy, and mean tardiness. The mean tardiness criterion, in particular, has been a standard way of measuring conformance to due dates, although it ignores the consequences of jobs completing early. However, this emphasis has changed with the current interest in Just in Time (JIT) production, which espouses the notion that earliness, as well as tardiness, should be discouraged. In JIT scheduling environment, jobs that complete early must be held in finished goods inventory until their due date, while jobs that complete after their due dates may cause a customer to shut down operations. Therefore, an ideal schedule is one in which all jobs finish exactly on their assigned due dates. (Baker & Scudder, 1990)). The concept of penalizing both earliness and tardiness has spawned a new and rapidly developing line of research in the scheduling field. Because the use of both earliness and tardiness penalties gives rise to a non regular performance measure, it has led to new methodological issues in the design of solution procedures.

Manuscript received March 30, 2015.

Abdelaziz Hamad, Department of Mathematics, College of Arts and Science, Raniah Branch, Taif University, Kingdom of Saudi Arabia

Mohamed Hassen Elzubair,Department of Mathematics, College of Arts and Science, Raniah Branch, Taif University, Kingdom of Saudi Arabia

II. Literature Review

The advent of Just-In-Time scheduling spawned a segment of the literature that investigated cost structures comprising both earliness costs and tardiness costs when processing times and due dates are given. The concept was introduced by Kanet (1981), who analyzed the minimization of total absolute deviation from a common due date, under the assumption that the due date is late enough. This objective is equivalent to an E/T problem in which the unit costs of earliness and tardiness are symmetric and the same for all jobs

Cheng & Gupta (1989) have conducted an extensive survey of scheduling research involving the due date determination decision. The problem of scheduling jobs to meet their due dates is both theoretically challenging and practically significant. That it is theoretically challenging is due to the nature of the scheduling problem itself, which, in the general as we mentioned is known to be a difficult combinatorial problem. On the practical side, since all companies use due date in one form or another, they are concerned about accurate due date quotation and their ability to meet the set due date. In this research we consider the problem of single machine with a set of simultaneously available jobs, each assigned a common or distinct due date, our objective is to sequence the jobs on the machines in an optimal fashion to minimize a penalty function. The penalty function depends on cost factors, which are related to the values of the individual job earliness and tardiness.

In the case of single machine with given distinct due date Garey and Wilfong (1988) were the first to show that this problem is NP-complete. Recent papers exploring solution procedures include Abdul-Razaq and Potts (1988), and Ow & Morton (1989. Abdul-Razaq and Potts (1988) solve this type of problem, with a due date penalty included, but they consider only schedules without inserted idle time. Their solution method is a branch-and bound scheme, and they use a relaxed dynamic programming procedure to obtain good bounds. Their computational results suggest that problems containing more that 20 jobs may lead to excessive solution times..

In their state of the art review of Earliness/Tardiness (E/T) scheduling, Baker and Scudder (1990) indicate that while common due date scheduling problems have been well researched, less progress has been made with the distinct due date scheduling problem. In general the single machine, distinct due date machine scheduling problem closely models the situation faced by Just-in-Time manufactures James and Buchanan (1997) considered the schedule with inserted idle time for small systems and have presented Tabu search algorithm with compressed solution space to determine a sub-optimal solution. Recently Nordin and Fatimah (2011) developed heuristic algorithm to find optimal solutions to minimized earliness and tardiness costs . In this paper we try to develop neural network model to solve the earliness and tardiness single machine. In this research, separate earliness and tardiness penalties are considered for each job with given due date. The nature of cost of early completion stems from additional inventory holding cost and ineffective use of resources. On the other hand, cost of tardy completion arises from penalty for late delivery and loss of goodwill. Therefore, it is relevant to consider the problem with separate earliness and tardiness penalty for each job.

III. Methodology

An artificial neural network is a collection of highly interconnected processing units that has the ability to learn and store patterns as well as to generalize when presented with new patterns. The 'learnt' information is stored in the form of numerical values, called weights that are assigned to the connections between the processing units of the network. A neural network usually consists of an input layer, one or more hidden layers and an output layer. Before the network is trained, the weights are assigned small, randomly determined values. Through a training procedure, such as back propagation, the network's weights are modified incrementally until the network is deemed to have learnt the relationship. This type of learning is a supervised type of learning. When a pattern is applied at the input layer, the stimulus is fed forward until final outputs are calculated at the output layer. The network's outputs are compared with the desired result for the pattern considered and the errors are computed. These errors are then propagated backwards through the network as feedback to the preceding layers to determine the changes in the connection weights to minimize the errors. A series of such input-output training examples is presented repeatedly until the total sum of the squares of these errors is reduced to an acceptable minimum. At this point the network is considered 'trained'. Data presented at the input layer of a trained network will result in values from the output layer consistent with the relationship learnt by the network from the training examples.

In this paper we need to develop a multi layer neural network model to finding solution to single machine with due date by using the vector representation of the early/ tardy scheduling problem. we divide the model according to the types of the due date into two cases. The first case is concerned with common due date earliness and tardiness scheduling problem and the second one with distinct due date . In both cases we shall consider the non-preemptive and non-delay scheduling which means there is no inserted idle time**.** We organize this paper as follows. In Section. 4 we determine the solution of the common due date earliness and tardiness and in Section 5 we discuss the solution of distinct due date earliness and tardiness.

IV. Single Machine with Common Due Date.

In this case a set of n independent jobs has to be scheduled on a single machine, which can handle only one job at a time. The machine is assumed to be continuously available from time 0 onwards. Job J_i (i = 1,…,n) has a given processing time P_i and should ideally be completed at a given common due date d. A generalization of the basic single machine model with earliness and tardiness penalties derived from the notion that each job has its own earliness and tardiness penalties. In particular, the objective function can be written as

$$f(S)=\sum_{i=1}^{n}(\alpha_i E_i+\beta_i T_i) \qquad (01)$$

where

S Schedule for the n jobs;

T_i max {0, C_i-d};

E_i max {0, d -C_i};

d common due date;

C_i time job i is completed after schedule S is started at time t = 0;

α_i Penalty per unit time for the earliness of job i;

β_i penalty per unit time for the tardiness of job i.

A. A Neural Network for Single Machine Common Due Date

The neural network that is proposed for the single machine common due date schedule problem is organized into three layers of processing units. There is an input layer of 10 units, a hidden layer, and an output layer that has a single unit. The number of units in the input and output layers is dictated by the specific representation adopted for the schedule problem. In the proposed representation, the input layer contains the information describing the problem in the form of a vector of continuous values. The 10 input units are designed to contain the following information for each of the n jobs that have to be scheduled:

$$unit1 = \frac{P_i}{M_p}, \quad (02)$$

$$unit2 = \frac{d}{100}, \quad (03)$$

$$unit3 = \frac{SL_i}{M_{SL}}, \quad (04)$$

$$unit4 = \frac{\alpha_i}{10.0}, \quad (05)$$

$$unit5 = \frac{\beta_i}{10.0} \quad (06)$$

$$unit6 = \frac{\overline{P}}{M_p}, \quad (07)$$

$$unit7 = 1, \quad (08)$$

$$unit8 = \frac{\overline{SL}}{M_{SL}}, \quad (09)$$

$$unit9 = \sqrt{\frac{\sum (P_i - \overline{P})^2}{n \times \overline{P}^2}}, \quad (10)$$

$$unit10 = \sqrt{\frac{\sum (Sl_i - \overline{SL})^2}{n \times \overline{SL}^2}}, \quad (11)$$

Where

P_i : Processing time of job i;

Sl_i : Slack of job i is the different between the due date and processing time $= d - P_i$;

M_P : Longest processing time among the n jobs = max $\{P_i\}$,

M_{SL} : Largest slack for the n jobs = max $\{Sl_i\}$, i $\in$n;

$$\overline{P} : \frac{\sum_{i=1}^{n} P_i}{n};$$

$$\overline{SL} : \frac{\sum_{i=1}^{n} SL_i}{n}.$$

The neural network is trained by presenting it with a predefined set of input and target output patterns. Each job is represented by 10 input vector, which holds information particular to that job and in relation to the other jobs in the problem. The output unit assumes values that are in the range of 0.10-0.90 (see equation 12), the magnitude being an indication of where the job represented at the input layer should desirably lie in the schedule. Low values suggest lead positions in the schedule; higher values indicate less priority and hence position towards the end of the schedule. The target associated with each input training pattern is a value that indicates the position occupied in the optimal schedule. The target value G_i for the job holding the i position in the optimal schedule is determined as

$$G_i = 0.1 + 0.8\left(\frac{i-1}{n-1}\right), \quad i = 1,\ldots,n. \quad (12)$$

Equation (12) ensures that the n target values are distributed evenly between 0.1– 0.9. The number of units in the hidden layer is selected by trial and error during the training phase. The final network for earliness and tardiness single machine with common due date has **8** units in its hidden layer and **1** unit of output layer, therefore known as **10-8-1** network configuration.

B. Numerical Example

Example 1

Now, the trained neural network is used to find the schedule for minimizing the cost function of equation.1. **Table 1** shows a 3-job single machine and common due date. The 3- jobs are converted first into their vector representation by using the set of equations (2-11). The result of this pre-processing stage is presented in **Table 2** where the vectors V_1-V_3 represented job numbers 1-3, respectively. To solve the schedule problem, each vector is presented individually at the input layer of the neural network. A feed forward procedure of calculations generates a value that appears at the output unit for each of the three input vectors. The output computed by the neural network for each of the input vectors is given in the right most column of Table 2

Table 1: 3-job scheduling problem

Job i	P_i	d	Sl_i	α_i	β_i
1	8	15	7	1	1
2	10	15	5	1	1
3	4	15	11	1	1

Table 2**: Problem representation for the example described in** Table 1

Input units

job	U_1	U_2	U_3	U_4	U_5	U_6	U_7	U_8	U_9	U_{10}	out
V_1	0.80	0.15	0.63	0.10	0.10	0.73	1.00	0.70	0.34	0.33	0.78
V_2	1.00	0.15	0.45	0.10	0.10	0.73	1.00	0.70	0.34	0.33	0.11

V_3	0.40	0.15	1.00	0.10	0.10	0.73	1.00	0.70	0.34	0.33	0.27

Scheduling the jobs in the order of the increasing output values results in the job schedule J_2- J_3- J_1 with total cost equal 13 units. The optimal schedule, on other hand, is J_2- J_3- J_1 with total cost equal 13 units. In this example the neural network scheduling give us optimal solution.

Example 2

In this example we use the same procedure as in example **1** The output computed by neural network professional software II for each of the input vectors is given in the right most column of **Table 4**

Table 3**:** 4-job scheduling problem

Job i	P_i	d	α_i	β_i	Sl_i
1	8	15	1	1	7
2	10	15	1	1	5
3	4	15	1	1	11
4	7	15	1	1	8

Table 4: Problem representation for the example described in Table 3

Input units

job	U_1	U_2	U_3	U_4	U_5	U_6	U_7	U_8	U_9	U_{10}	out
V_1	0.80	0.15	0.63	0.10	0.10	0.73	1.00	0.70	0.30	0.28	0.75
V_2	1.00	0.15	0.45	0.10	0.10	0.73	1.00	0.70	0.30	0.28	0.10
V_3	0.40	0.15	1.00	0.10	0.10	0.73	1.00	0.70	0.30	0.28	0.27
V_4	0.70	0.15	0.72	0.10	0.10	0.73	1.00	0.70	0.34	0.28	0.42

Scheduling the jobs in the order of the increasing output values results in the job schedule J_2- J_3- J_4-J_1 with total cost equal 26 units. The optimal schedule, on other hand, is J_1- J_4-J_3-J_2 with total cost equal 25 units. That means the neural network scheduling in this example gives us sub optimal solution.

V. Single Machine with Distinct Due Date

In this Section a set of n independent jobs has to be scheduled on a single machine, which can handle only one job at a time. The machine is assumed to be continuously available from time 0 onwards. Job J_i (i= 1,…,n) has a given processing time P_i and should ideally be completed at a given distinct due date d_i.

The general earliness and tardiness model has different due dates in the job set. This feature tends to make it more difficult to determine a minimum cost schedule than in the problem of common due date discussed thus far. However, if the due dates are treated as decision variables, the problem turns out to be relatively simple. In the remainder of this Section we assume that the due dates are given and distinct and the slack for each job is small, and the objective function is Garey, Tarjan & Wilfong (1988) were the first to show that this problem is NP-complete. This means that the solutions are mostly provided in the form of heuristics and thus do not provide a reasonable generalization to the problem.

A. A Neural Network For Single Machine Distinct Due Date

$$f(S)=\sum_{i=1}^{n}(\alpha_i E_i+\beta_i T_i). \qquad (13)$$

The neural network that is proposed for the single machine distinct due date schedule problem is organized into three layers of processing units. There is an input layer of 11 units, a hidden layer, and an output layer that has a single unit. The number of units in the input and output layers is dictated by the specific representation adopted for the schedule problem. In the proposed representation, the input layer contains the information describing the problem in the form of a vector of continuous values. The eleven input units are designed to contain the following information for each of the n jobs that have to be scheduled

where

d_i **:**Distinct due date for job i;

Sl_i : Slack is the different between the due date and processing time $= d - P_i$;

$$unit1 = \frac{P_i}{M_p}, \quad (14)$$

$$unit2 = \frac{d_i}{M_d}, \quad (15)$$

$$unit3 = \frac{SL_i}{M_{sl}}, \quad (16)$$

$$unit4 = \frac{\alpha_i}{10.0}, \quad (17)$$

$$unit5 = \frac{\beta_i}{10.0} \quad (18)$$

$$unit6 = \frac{\overline{p}}{M_p}, \quad (19)$$

$$unit7 = \frac{\overline{d}}{M_d}, \quad (20)$$

$$unit8 = \frac{\overline{SL}}{M_{sl}}, \quad (21)$$

$$unit9 = \sqrt{\frac{\sum (P_i - \overline{P})^2}{n \times \overline{P}^2}}, \quad (22)$$

$$unit10 = \sqrt{\frac{\sum (d_i - \overline{d})^2}{n \times \overline{d}^2}}, \quad (23)$$

$$unit11 = \sqrt{\frac{\sum (Sl_i - \overline{SL})^2}{n \times \overline{SL}^2}}, \quad (24)$$

M_p : longest processing time among the n jobs = max $\{P_i\}$;
M_d : latest due date of the n jobs = max $\{d_i\}$, $i \in n$;
M_{SL} : largest slack for the n jobs = max $\{SL_i\}$, $i \in n$;

$$\overline{P} : \frac{\sum_{i=1}^{n} P_i}{n};$$

$$\overline{SL} : \frac{\sum_{i=1}^{n} SL_i}{n};$$

$$\overline{d} : \frac{\sum_{i=1}^{n} d_i}{n}.$$

The neural network is trained by presenting it with a predefined set of input and target output patterns. Each job is represented by a 11-input vector, which holds information particular to that job and in relation to the other jobs in the problem. The output unit assumes values that are in the range of 0.20-0.90, the magnitude being an indication of where the job represented at the input layer should desirably lie in the schedule. Low values suggest lead positions in the schedule; higher values indicate less priority and hence position towards the end of the schedule. The target associated with each input training pattern is a value that indicates the position occupied in the optimal schedule. The target value G_i for the job holding the *ith* position in the optimal schedule is determined as

$$G_i = 0.2 + 0.7\left(\frac{i-1}{n-1}\right), \qquad i = 1,..,n. \quad (25)$$

Equation (25) ensures that the n target values are distributed evenly between 0.2– 0.9. The number of units in the hidden layer is selected by trial and error during the training phase. The final network for single machine with distinct due date has **9** units in its hidden layer and **1** unit of output layer, therefore known as **11-9-1** network configuration..

B. Numerical Example

The trained neural network is used to find the schedule for minimizing the cost function of equation (13). **Table 5** shows 8-job single machine tardiness and earliness problem. The eight jobs are converted first into their vector representation by using the set of equations (14-24). The result of this pre-processing stage is presented in **Table 5**., where the vectors V_1-V_8 represented job numbers 1-8, respectively. To solve the schedule problem, each vector is presented individually at the input layer of the neural network. A feed forward procedure of calculations generates a value that appears at the output unit for each of the eight input vectors. The output computed by the neural network for each of the input vectors is given in the right most column of **Table 6**

Table 5: 8- job scheduling problem

Job i	P_i	d_i	α_i	β_i	SL_i
1	125	278	3	7	153
2	125	291	5	7	166
3	121	395	2	7	274
4	102	336	4	6	234
5	58	399	2	8	341
6	124	262	5	8	138
7	193	259	1	6	66
8	68	329	2	7	261

Table 6: Problem representation for the example described in Table 5

Input units

job	U_1	U_2	U_3	U_4	U_5	U_6	U_7	U_8	U_9	U_{10}	U_{11}	Out
V_1	0.65	0.70	0.45	0.30	0.70	0.95	0.94	0.60	0.34	0.19	0.41	0.52

V_2	0.65	0.73	0.49	0.50	0.70	0.95	0.94	0.60	0.34	0.19	0.41	0.45
V_3	0.63	0.99	0.80	0.20	0.70	0.95	0.94	0.60	0.34	0.19	0.41	0.56
V_4	0.53	0.84	0.69	0.40	0.60	0.95	0.94	0.60	0.34	0.19	0.41	0.40
V_5	0.30	1.00	1.00	0.20	0.80	0.95	0.94	0.60	0.34	0.19	0.41	0.23
V_6	0.64	0.66	0.40	0.50	0.80	0.95	0.94	0.60	0.34	0.19	0.41	0.42
V_7	1.00	0.65	0.19	0.10	0.60	0.95	0.94	0.60	0.34	0.19	0.41	0.81
V_8	0.35	0.82	0.77	0.20	0.70	0.95	0.94	0.60	0.34	0.19	0.41	0.27

Scheduling the jobs in the order of the increasing output values results in the job schedule, J_5- J_8- J_4- J_6- J_2- J_1- J_3- J_7. with total cost of 12048 units

VI. Conclusions

In this paper we have studied the solutions of the earliness and tardiness single machine criteria with back propagation neural network . In order to obtain the solution of the problem, the model of the problem has been divided into two cases with common and distinct due date. In both cases the back propagation neural network was used to find the optimal or near optimal solutions. It is found that the back propagation neural network for common due date has 10-8-1 configuration while for distinct due date has 11-9-1 configuration . The back propagation neural network give us optimal or near optimal solutions for the two cases . Further research can be carry out for multi machine

Acknowledgement

The authors would like to thanks University of Taif, Kingdom of Saudi Arabia for their financial support grant No. 1/435/3456.

References.

[1] Abdelaziz Hamad (2002) Application of neural network for solving job shop scheduling. PhD thesis : UTM, Malaysia

[2] Abdelaziz Hamad and Bahrom Sanugi (2011) Neural network and Scheduling Lambert Academic publication. Germany.

[3] Abdul-Razaq, T. & Potts, C. (1988). " Dynamic Programming State –Space Relaxation for Single Machine Scheduling." Journal of Operational Research Society. 39; 141-152.

[4] Aho, A. Hopcroft, J.E. and Ullman, J.D. (1974). Design and Analysis of Computer Algorithm. MA. : Addison-Wesley.

[5] Baker, K. and Scudder, G. (1990). "Sequencing with Earliness and Tardiness Penalties: A review." Operations Research. 38; 22 - 36.

[6] Cheng, T. C.E. and Gupta, M.C. (1989). "Survey of Scheduling Research Involving Due Date Determination Decisions." European Journal of Operational Research.38; 156-166

[7] El-Bouri, A., Balakrishnan, S. and Popplewell, N. (2000). " Sequencing Jobs on a Single Machine: A Neural Network Approach". European Journal of Operational Research. 126; 474-490.

[8] James, R. J. and Buchanan, J.T. (1997). " A Neighborhood Scheme with a Compressed Solution Space for the Early/Tardy Scheduling." European Journal of Operation Research. 102; 513-527.

[9] Garey, M., R. Tarjan and G. Wilfong (1988) One-Processor Scheduling with Symmetric Earliness and Tardiness Penalties. Mathematics of Operations Research 13, 330-348.

[10] James, R. and J. Buchanan (1997) A neighbourhood scheme with a compressed solution space for the early/tardy scheduling problem. European Journal of Operational Research 102, 513-527.

[11] Kanet, J. (1981) Minimizing the average deviation of job completion times about a common due date. *Naval Research Logistics Quarterly* 28, 643-651.

[12] M 'Hallah, R. (2007) Minimizing total earliness and tardiness on a single machine using a hybrid heuristic. *Computers & Operations Research* 34, 3126 – 3142.

[13] Ow, P. and T. Morton (1989) The single machine early/tardy problem. Management Science 35,177-191.