CHAPTER-V

ASSIGNMENT PROBLEM HEURISTIC ALGORITHMTO MINIMIZE MAKESPAN ON NON-IDENTICALPARALLEL MACHINES

The main results of this chapter have been published as detailed below:

Assignment Problem Heuristic Algorithm to Minimize Makespan on Non-Identical Parallel Machines.

Bulletin of Society for Mathematical Services & Standards (BSOMASS), Vol. No. 1 (2012) pp. 11-17.

5.1 INTRODUCTION

In this chapter we consider the problem of scheduling 'n' single operation jobs with a common due date on 'm' parallel machines so as to minimize the sum of the absolute lateness. In this case of non-identical machines we reduced the problem to a transportation problem that can be solved by a polynomial time algorithm. Furthermore we have given assignment problem heuristic algorithm to minimize makespan among all schedules that minimize the absolute lateness problem and to find the value of processing time and due date.

The study of the following single operation n job scheduling problem. All n jobs are available at time zero. Each job can be processed on any one of the non-identical machines in parallel. In this chapter we study transportation problem deals with the transportation of a single product from several sources to several sinks (destinations or demand). In general study, let there be m sources $S_1, S_2, ..., S_m$ with $a_i (i = 1, 2, ..., m)$ available supplies or capacity at each source i, to be allocated among n destination j. Let C_{ij} be the cost of shipping one unit from source i to destination j for each rout. Then it x_{ij} be the unites shipping per route from source i

to destination j, the problem is to determine the transportation schedule so as to minimize the total transportation cost satisfying the supply and demand conditions. The assignment problem is a special case of the transportation problem in which the objective is to assign a number of origins to the equal number of destinations at a minimum cost (maximum profit). Let d be the common due date (to be determined) and let $C_{[ij]}$, be the completion time on machine i for a job in position j. Let n_i represent the number of jobs scheduled on machine i. Let S be a given schedule and d be a given due date. The given schedule S consists of $S_1, S_2, ..., S_i, ..., S_m$ where S_i is the sequence on machine i. The total penalty function $f(S,d)$ is given by

$$f(d,S) = \sum_{i=1}^{m} \sum_{j=1}^{n_i} \left| C_{ij} - d \right|$$

The objective is to find a due date d^* and a schedule S^* such that $f(d,S)$ is minimized. The term on the right-hand side gives the sum of absolute lateness or the sum of absolute deviation of completion times about a common due date. We can define earliness and tardiness represented respectively by

$$E_{[ij]} = \max\left\{0,\ d - C_{ij}\right\}$$

and $T_{[ij]} = \max\{0, C_{ij} - d\}$.

The same objective function can now be written as

$$f(d,S) = \sum_{i=1}^{m}\sum_{j=1}^{n_i}(E_{ij} + T_{ij}).$$

This objective function can now be considered as the sum of the earliness and tardiness penalties.

This chapter is motivated by many earlier studies involving a single operation sequencing problem for n jobs with performance measure given by $f(d,S)$.

5.2 PREVIOUS RESULTS

For $m = 1$, the proposed problem reduces to a special case of the problem proposed by **Panwalkar et al**. [10] included due date, earliness and tardiness per unit penalties in their chapter. For the single machine case, schedule S consists of only one sequence. From that paper, we know the following.

- ***Rule 1***. For any specified sequence S, there exists an optimal value of d which coincides with a completion time of a job r in S.
- ***Rule 2***. The penalty function $f(d,S)$ can be written as

$$f(d,S)=(0)t_{[1]}+1t_{[2]}=2t_{[3]}+...+(r-1)t_{[r]}+(n-r)t_{[r+1]}+...+2t_{[n-1]}+1t_{[n]}$$

Thus, the positional penalties for sequence positions $1,2,3,...,n-1$ *and* $0,1,2,3,...$ for non-tardy jobs and $\ldots,3,2,1,$ for the tardy jobs and there exists an optimal sequence such that, when n is even, $n/2$ jobs will be non-tardy and $n/2$ jobs will be tardy (maximum positional penalty will be $n/2$). If n is odd, the non-tardy set will contain one more job than the tardy set (maximum positional penalty will be $(n-1/2)$.

- ***Rule 3.*** An optimal sequence can be found by assigning the largest job to the position with the smallest penalty (position 1), the second largest job in a position with the next larger penalty (position 2 or position n) and so on. Finally the optimal due date for the sequence will coincide with the completion time of the set of non-tardy jobs. Once an optimal sequence is found the schedule starts at time $R=0$.
- ***Rule 4.*** The optimal sequence thus obtained is V shaped.

Another way to analyses the above problem for $m=1$ is to consider it as an assignment problem of size $n\times n$. If $\lambda=(\lambda_1,\lambda_2,...,\lambda_n)$

represents a row vector of positive processing times, the cost elements of the assignment matrix will be as follows.

$$Q = \begin{bmatrix} 0_\lambda \\ 1_\lambda \\ . \\ k_\lambda \\ . \\ . \\ l_\lambda \\ . \\ 2_\lambda \\ 1_\lambda \end{bmatrix}$$

The first half rows represent (positional penalties) x (processing times for non-tardy jobs) and the bottom half of the matrix corresponds to penalties for tardy jobs. The $(kj)^{th}$ element of Q represents the contribution of job j to total cost if it is scheduled in the $(k+1)^{th}$ position and if that job is non-tardy for $k = 0,1,2,...$ The $(lj)^{th}$ element of Q represents the contribution of job j to total cost if it is tardy and scheduled in the $l th$ position from the end for $l = 1,2,...$ The solution of the assignment problem gives an optimal sequence and the same value of total penalty. Rows of the above matrix can be rearranged so that the positional penalties used for row 1 through n are in non-decreasing order (0, 1, 1, 2, 2...)

5.3 MINIMIZING TOTAL ABSOLUTE DEVIATION FROM A COMMON DUE-DATE

An important study on special case in the family of E/T problems involves minimizing the sum of absolute deviations of the job completion times from a common due date. **Sen** and **Gupta** [1] survey scheduling models involving due dates, but they cite only a handful of articles on the E/T model and obviously do not cover papers written after 1984. There are two ways to assign each pair of jobs that must be split between the two sets A and B; if n is even, we can create a fictitious additional job with zero processing time to complete the last pair. Therefore, as observed by **Bagchi, Sullivan** and **Chang** [2], the total number of optimal schedules is 2^r where

$$r = (n-1/2) \quad \text{if } n \text{ is odd}$$

$$= n/2 \quad \text{if } n \text{ is even}$$

Actually, this observation assumes that the processing times are unique. If there are ties, the number of optimal schedules is even greater. **J. Kanet** [3] we study Kanet's model to m non-identical machines. We reduce the problems to a transportation problem that can be solved by a polynomial time algorithm. A similar procedure has been given by **Bruno et al.** [4] for minimizing mean finishing time in m non-identical machines. We also consider the problem of

minimizing makespan in the case of m identical machines. This minimization will be over all optimal solutions of the absolute lateness problem. We present a heuristic algorithm for this problem**.** The model of Panwalkar et al. in a general case (when per unit penalties for due date, earliness and tardiness of a job are included) has been extended to m identical machines in parallel by **Cheng** [5]. Cheng implicitly assumes that the start time of the schedule in any machine must be at $R=0$ and he gave a heuristic algorithm. **Diamond** and **Cheng** [6] showed that the problem is NP-hard. **Sequencing books** [7, 8, and 9] also discuss many of these problems.

5.4 ANALYSIS OF THE PROPOSED MODEL

In the following we concentrate on the model of **Panwalkar et al.** [10] but the results are easily applicable to Kanet's model. Since the processing times of a job on different machines may be different, we will use λ_{ij} to denote the processing time required on machine $i\,(1 \le i \le m)$ for job $j\,(1 \le i \le n)$. The $m-$row, $n-$column processing time matrix will be denoted by λ . All processing times are assumed to be positive. Let $S=\{S_1, S_2, ..., S_m\}$ denote a schedule and let n_i

denote the number of jobs scheduled on machine i. Then $0 \le n_i \le n$ for all i and $\sum_{i=1}^{m} n_i = n$.

Property 1. *For any given schedule* S, *an optimum due date will coincide with completion time of some job on one of the machines.*

Proof: For any specific machine this result follows from **Rule 1.** Let $S = (S_1,...,S_m)$ be a schedule for m machines and let $d = (d_1,...,d_m)$ where d_i is the optimal due date for the set of jobs scheduled on machine $i, (i = 1,...,m)$. Now the proof follows from the fact that we can enlarge the due date and, accordingly, the start time of the schedule for any given machine without affecting the optimum value of the total cost.

Property 2. *Positional penalties for the non-tardy jobs on any given machine are 0, 1, 2, 3 … and for tardy jobs are ..., 3, 2, 1. Therefore assuming n > m, there exists an optimal sequence in which each machine will process at least one job (also implying that the maximum number of jobs processed on any machine will be* $n - m + 1$ *). If* n_i, *the number of jobs scheduled on machine i is even, half the jobs will be non-tardy and half the jobs will be tardy. For the odd*

value of n_i*, the non-tardy set will contain one more job than the tardy set.*

Proof. **By Rule 1** for each specific machine there is a job r that will be completed at the due date d in an optimal order, and by **Rule 2,** the coefficients of the non-tardy jobs for this machine will be 0, 1, 2… and for tardy jobs will be ... 3, 2, 1. Now, it is obvious that assuming there are more jobs than machines, all the first positions of the machines will be filled first. If n_i is the number of jobs scheduled on machine i, then by Rule 3 half the jobs will be non-tardy and half will be tardy in the case of n_i being even. And for the odd value of n_i the non-tardy set will contain one more job than the tardy set. Now define the following $m(n-m+1)\times n$ matrix Q as,

$$Q=\begin{bmatrix} 0_\lambda \\ 1_\lambda \\ . \\ k_\lambda \\ . \\ . \\ l_\lambda \\ . \\ 2_\lambda \\ 1_\lambda \end{bmatrix}$$

where $k = 0, 1, 2, ..., \frac{n-m+1}{2}$, $l = 1, 2, ..., \frac{n-m+1}{2}$ if $n-m+1$ is even and $k = 0, 1, 2 ..., \left[\frac{n-m+1}{2}\right], l = 1, 2, ..., \left[\frac{n-m+1}{2}\right]$ if $n-m+1$ is odd and define $[x]$ to be the smallest integer greater than or equal to x and $[x]$ the greatest integer smaller than or equal to x.

5.5 SCHEDULING NON-IDENTICAL PARALLEL MACHINES

The scheduling of independent jobs on non-identical parallel machines in order to minimize "*makespan*". There are n independent jobs, each of which have a definite processing time and are allowed to be processed on any of the m non-identical parallel machines. Each machine has a different velocity denoted by V_i. In scheduling problem, *makespan* $(C_{\max})$ is equivalent to the completion time of the last job leaving the system. The small $C_{\max}$ usually implies a high utilization. Therefore, reducing the $C_{\max}$ should also lead to a higher throughput rate (**Kashan, et al.** 2008). In this study, the *genetic algorithm (*GA*)* has been applied to solve this problem. Scheduling is an essential function in production management. Scheduling determines what is going to be made, when, where and with what resources (**Cardon, et al.** 2000). The production

scheduling is an important decision making in operational level and it is a difficult problem depending on the number of calculations required to obtain a scheduling that optimizes the chosen criterion. In modern manufacturing environment, many scheduling problems arise. In the factory, depending on machine layout and job flow, several kinds of shop exist (**Moon, et al.** 2002). *A* flow shop is a shop in which machines are arranged in series; jobs begin processing on an initial machine, proceed through several intermediary machines and conclude on a final machine. In a job shop, jobs can be processed on machines in any order. In other words, specific machines order restriction is not imposed. There are m machines and n jobs to be processed; each job requires l operations that are to be processed in order. Jobs may not require all m machines and they may have to visit some machines more than once. There is only one machine that can perform a given operation. In practice, there are often multiple copies of the same machine (parallel machine).

ASSIGNMENT ALGORITHM 1

Various steps of the computational procedure for obtaining an optimum assignment may be summarized as follows;

- **Step 1.** Check whether the number of rows and columns in the cost matrix are equal. If not, add dummy rows (columns) to form a square matrix.
- **Step 2**: In the square cost matrix, reduce each row and columns element by the lowest element of that row and columns.
- **Step 3:** Examine the rows and columns successively until a row and column with exactly single zeros in a particular row and column, choose arbitrarily any one of these and cross out all other zeros of that row and column.
- **Step 4:** If each row and each column has one and only one marked zero, the optimum allocation is attained which is indicated by the marked position, otherwise go to next step.
- **Step 5**: Draw straight lines through all unticked rows and ticked columns.
- **Step 6:** If the minimum number of lines passing through all the zeros is equal to the number of rows or columns, the optimum solution is attained by an arbitrary allocation in the positions of the zeros not crossed in step 3. Otherwise go to next step.

- **Step 7**: The smallest element not covered by any of the lines of step 5. Subtract this form all the uncrossed elements and add the same at the point of intersection of the two lines and other elements crossed by the lines remain unchanged.
- **Step 8**: Go to step 5 and repeat the procedure till an optimum solution is attained.

➢ **ALGORITHM 2**

Algorithm 2 (optimal solution for sum of the absolute lateness)

- **Step 1:** Solve the $m(n-m+1)$ row, n column assignment problem indicated in property 3 above to determine jobs to be scheduled on individual machines.
- **Step 2:** Assuming that the first job on machine i is scheduled at time zero, calculate the completion time of the last non-tardy job and set di equal to that time. Set common due date d^* to the maximum value of $(i=1,2,...,m)$ and move the starting time of the first job on machine i to $d^* - d_i$.

Note: If the model to be solved is Kanet's model there is a large value of due date d given, thus the algorithm consists of Step 1 and Step 2' as follows:

- **Step 2':** Assuming that the first job on machine i is scheduled at time zero, calculate the completion time of the last non-tardy job and set di equal to that time, and move the starting time of the first job on machine i to $d^* - d_i$.

5.6 A NUMERICAL EXAMPLE

Consider the following two machines six job problem. Processing time matrix λ is as follows:

$$\begin{bmatrix} 14 & 12 & 10 & 6 & 4 & 6 \\ 10 & 12 & 14 & 16 & 18 & 10 \end{bmatrix}$$

The assignment matrix will have 6 rows and 6 columns as shown. Asterisks indicate an optimal solution to the assignment problem.

$$\begin{bmatrix} 0^* & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0^* & 0 & 0 & 0 \\ 14 & 12 & 10 & 6 & 4^* & 6 \\ 10 & 12^* & 14 & 16 & 18 & 10 \\ 14 & 12 & 10 & 6^* & 4 & 6 \\ 10 & 12 & 14 & 16 & 18 & 10^* \end{bmatrix}$$

From the solution we find that machine 1 will process job 1, 5 and 4; and machine 2 will process job 3, 2, 6. Calculation for due dates d_i are:

Machine 1, sequences 1, 5, 4 $\quad d_1$=18

Machine 2, sequences 3, 2, 6 $\quad d_2$=26

An optimal due date is therefore 26. Complete schedules S as follows, see Figure 1. Jobs are started at time zero on machine 1 (sequence 1, 5, 4 with two non-tardy jobs). Jobs are started at time 10 on machine 2 (sequence 3, 2, and 6 with two non-tardy jobs). Total penalty is 32. The total optimum assignment schedule is: A→1, B→3, C→5, D→2, E→4, F→6.

Although the value of the objective function is the same

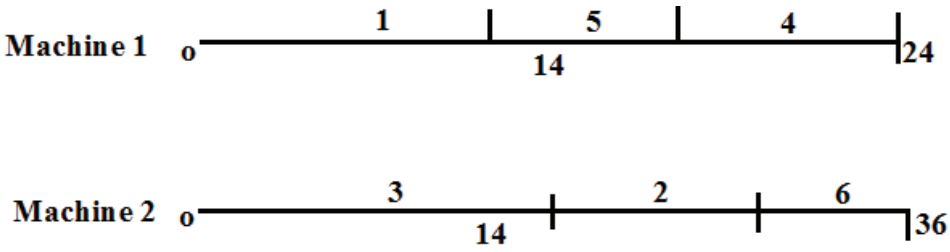

Fig.1. Schedule for the numerical example.

5.7 CONCLUSION

In the case of non-identical machines we reduced the problem to a transportation problem that can be solved by a polynomial time algorithm. Furthermore we have given assignment problem heuristic algorithm to minimize makespan among all schedules that minimize the absolute lateness problem and to find the value of processing time and due date. The assignment problem in witch objective is to assign a number of origins to the equal number of destinations at a minimum cost (maximum profit). We define the total optimum assignment schedule, optimal due date and total penalty.

5.8 REFERENCES

1. **Sen, T., and Gupta, S.:** A State-of-the-Art Survey of Static Scheduling Research Involving Due Dates. *OMEGA* 12, No. 1, pp. 62-76, 1984.
2. **Bagchi, U. R., Sullivan.,** and **Chang, Y.:** *Minimizing Mean Absolute Deviation of Completion Times about a Common Due Date*. Naval Res. Logist. Quart. 33, pp. 227-240, 1986.
3. **Kanet, J.:** Minimizing the Average Deviation of Completion Times About a Common Due Date. *Naval Res. Logist.* Q. 28, pp. 643-651**,** 1981.
4. **Bruno, J., Coffman, E. G. and Sethi.:** Scheduling Independent Tasks to Reduce Mean Finishing-Time. *Comm. ACM* 17, pp. 382-387, 1974.
5. **Cheng, T. C. E.:** A Heuristic for Common Due-Date Assignment and Job Scheduling on Parallel Machines. *J. Opl Res. Soc.* 40, pp. 1129-1135, 1989.
6. **Diamond, J. E., and Cheng, T. C. E.:** Approximation Solution and Error Bounds for Common Due-Date Assignment and Job Scheduling on Parallel Machines. *(Department of Acturial and Management Science, University of Manitoba),* 1991.
7. **Baker, K. R.:** Introduction to Sequencing and Scheduling. *Wiley, New York,* 1974.
8. **Bruno, J.:** Mean Weighted Flow-time Criterion. In Computer and Job-Shop Scheduling Theory *(E. G. COFFMAN, Jr. Ed.),* pp, 101-137. *Wiley, New York,* 1976.

9. **French, S.:** Sequencing and Scheduling. *Wiley, New York,* 1982.

10. **Panwalker, S., Smith, M. and Seidmann, A.:** Common Due Date Assignment to Minimize Total Penalty for the One Machine Scheduling Problem. *Opns Res.* 30, pp. 391-399, 1982.