CHAPTER-IV

COMMON DUE-DATE ASSIGNMENT AND JOB SCHEDULING ON SINGLE MACHINE AND PARALLEL MACHINES

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

Common Due-Date Assignment and Job Scheduling on Single Machine and Parallel Machines.

Bulletion of Society for Mathematical Services & Standards (BSOMASS), Vol. I No. 1 (2012), PP. 27-37.

4.1 INTRODUCTION

In this chapter we consider a total penalty for the n job, one machine scheduling problem in which all jobs have a common due date. This penalty function is based on the due date value and on the earliness or the lateness of each job in the selected sequence. The main objective is to determine the optimal value of this due date and an optimal sequence to minimize a total penalty function. We prove that the optimal due date result can be generalized to the parallel machine problem. The problem of simultaneously available jobs on several parallel and identical machines. The problem is to find the optimal due date, assuming this to be the same for all jobs and we present a simple heuristic to find an approximate solution. On the basis of a limited experiment, we observe that the heuristic is very effective solution.

We consider the basic one machine sequencing problem under the common assumptions listed in **Baker** [1]. All jobs have a common (but unknown) due date. The objective is to find an optimal value of the due date and an optimal sequence which minimizes the total penalty based on the due date value and the earliness or tardiness of each job. The few studies analyzing due date selection procedures

that of **Baker** and **Bertrand** [2] are the most recent and directly related to the present one. They considered (among other things) a static one machine problem. One of the due date assignment procedures involved in their study was the common due date.

In this chapter, we consider the basic problem of parallel-machine scheduling under stable conditions, which are characterized by the assumptions discussed in detail by various authors for example, **Conway et al.** [3], **RinnooyKan** [4] and **French** [5]. Given a set of simultaneously available jobs, each assigned a common due-date; our objective is to find the optimal due-date and to sequence the jobs on the machines in an optimal fashion to minimize a penalty function. The penalty function depends on several relevant cost factors which are related to the values of the assigned due-date, as well as the individual job earliness and tardiness values.

The problem of optimal assignment of a common due-date and sequencing n jobs on a single machine has been considered by **Panwalkar et al.** [6] they show that the optimal common due-date value must coincide with one of the job completion times. They also present an efficient scheduling algorithm to solve the sequencing problem.

Most studies involving scheduling with due dates have treated due dates as exogenous decisions. Other job shop studies using various due dates have been reported in the literature (see for example**, Eilon** and **Chaudhary** [7], **Jones** [8], **Weeks** [9], and **Weeks** and **Fryer** [10]). Some of these studies include specified common due dates for the one machine problem with minimum total tardiness criterion.

In this chapter (due date, earliness, and tardiness) are all linear. Linear cost functions present a case that is more tractable than that occurring with nonlinear costs. Two of the costs, due date and tardiness, are related to customer objectives while the earliness cost can be viewed as an important element that concerns the production shop and also present a simple proof of the optimality result for the single-machine problem, and then generalize the result to the parallel-machine problem. While in the single-machine case the job-sequencing problem can easily be solved, this problem is too complex to be solved analytically under the parallel-machine situation. We therefore propose a simple heuristic to help determine a near-optimal solution to such a problem.

The due date assignment problems make practical sense when a firm offers a due date to its customers during sale negotiations and has to offer a price reduction when the due date is far away from the

expected one. In many instances, due dates are negotiated rather than simply dictated by the customers. The later the due dates are fixed, the higher the probability that the product will be completed or delivered on time. In order to maintain a good image among the customers, many companies tolerate reasonable holding costs in favor of keeping the established due dates. Thus, the decision maker has to balance the losses resulting from the holding costs and the advantages of fulfilling the orders in time.

Their objective, however, was only to find the lowest value of due date such that no job would be tardy**.**

4.2 FORMULATION OF THE PROBLEM

Let N= {1, 2,…, n} be a set of n independent jobs and M a set of m parallel and identical machines. Job i requires t_i amount of processing time on one of the machine, $\forall i \in N$. We assume that the job labeling in such that $t_1 \le t_2 \le ..... \le t_n$. i.e., sequence 1, 2,...,n represents the SPT sequence. All jobs are assumed to have a common due date d.

Let Π be the set of all possible job sequences. Also let σ denote an arbitrary member of Π, and the subscript j[i] denote the job in position [i] on machine j. $\forall [i] \in N, \forall j \in M$, in sequence σ. It

$C_{j[i]}, E_{j[i]}$ and $T_{j[i]}$ denote the completion time, earliness and tardiness of job $j[i]$ in σ, then

$E_{j[i]} = \max(0, d - C_{j[i]})$ And $T_{j[i]} = \max(0, C_{j[i]} - d)$

Our objective is to find the optimal common due-date, d^* and the optimal job sequence, σ^* such that some jobs are early and others are late so as to minimize a penalty function given by

$$f(d,\sigma) = \sum_{j\in M}\sum_{i\in N}\{P_1 d + P_2 E_{j[i]} + P_3 T_{j[i]}\} \quad (4.2.1)$$

$$= nP_1 d + P_2\sum_{i\in N}(d - C_{[i]}) + P_3\sum(C_{[i]} - d) \quad (4.2.2)$$

where P_1, P_2 and P_3 are non-negative constants representing the due-date assignment cost per unit time, the earliness and tardiness costs per unit time, respectively.

4.3 PRELIMINARY ANALYSIS

We now prove the following two lemmas related to the optimal due date value. Throughout the reminders of this section we assume $P_1 < P_3$.

Lemma1. *For any specified sequence σ, there exists an optimal value of d which coincides with the completion time of one of the jobs in σ.*

Proof. Suppose $d < C_{[1]}$. Then all jobs are tardy. If the due date is increased to $C_{[1]}$ the total penalty will increase by $n(P_1 \geq P_3)(C_{[1]} - d)$. This quantity is non-positive.

Now suppose

$C_{[i]} < d < C_{[i+1]},\ for\ \ i = 1,2,...,n-1.$

Let $x = d - C_{[i]}$ and Let $\gamma = C_{[i+1]} - d$ so that $x > 0$ and $y > 0$.

If the due date is changed to $C_{[i]}$, the new penalty is given by

$$f(C_{[i]},\sigma) = f(d,\sigma) + x(n(P_3 - P_1) - i(P_2 + P_3))$$

Similarly, if the due date is changed to $C_{[i+1]}$, the penalty is given by

$$f(C_{[i+1]},\sigma) = f(d,\sigma) + y(n(P_3 - P_1) - i(P_2 + P_3))$$

Obviously, $f(C_{[i]},\sigma) \leq f(d,\sigma)$ if $(n(P_3 - P_1) - i(P_2 + P_3)) \leq 0$ and $f(C_{[i+1]},\sigma) < f(d,\sigma)$ otherwise.

It is now clear that an optimal value of d concedes with the completion time of a job.

Lemma 2. *For any specified sequence σ. There exists an optimal due date equal to $C_{[k]}$, where k is the smallest integral value greater than or equal to $n(P_3 - P_1)/(P_2 + P_3)$.*

Proof. From lemma 1 we know that an optimal due date value can coincide with the completion time of some job. Let this be the job in position k where$1 \le k \le n$.

The total penalty is then equal to $f(C_{[k]},\sigma)$ since this due date is optimal, we know that for any$\Delta > 0$.

$f(C_{[k]}+\Delta,\sigma) - f(C_{[k]},\sigma) \ge 0$

The right shift of the due date from $C_{[k]}$ to $C_{[k]}+\Delta$ causes an increase in the first two components of the total penalty by at least $\Delta(nP_1+KP_2)$ and causes a decrease in the third compound by no more than $\Delta(n-K)P_3$.

Therefore, $\Delta(nP_1+KP_2-(n-k)P_3) \ge 0$

or $K \ge n(P_3-P_1)/(P_2+P_3)\ldots$ (4.3.1)

Also $f(C_{[k]}-\Delta,\sigma) - f(C_{[k]},\sigma) \ge 0$,

or, the left shift of the due date results in

$$K-1 \le n(P_3-P_1)/(P_2+P_3)$$

Since K is an integer, the proof of lemma 2 follows immediately.

From the discussion thus for, it is clear that for any sequence, exactly k jobs will be non-tardy$(K=0 \ \ if \ P_1 \ge P_3)$.

The total penalty is equal to

$$f(C_{[k]},\sigma)=\sum_{i=1}^{n}(P_1 d+P_2 E_{[i]}+P_3 T_{[i]}).$$

Substituting $C_{[k]}=t_{[1]}+t_{[2]}+t_{[3]}+.....+t_{[k]}$ and simplifying, we get,

$$f(C_{[k]},\sigma)=\sum_{j=1}^{k}(nP_1+(j-1)P_2 t_{[j]}+\sum_{j=k+1}^{n}P_3(n+1-j)t_{[j]})$$

$$=f(C_{[k]},\sigma)=\sum_{j=1}^{k}\gamma_j t_{[j]},$$

where γ_j is equal to $nP_1+(j-1)P_2$.

For $j\le k$ and is equal to $P_3(n+1-j)$ for $j>k$.

To find the optimal sequence, we need to find the minimal penalty among all $\sigma\in\pi$. The term γ_j may be viewed as the positional penalty for sequence position j.

Lemma 3. *Quantity* $\left(\sum_{j=1}^{n}\gamma_j t_{[j]}\right)$ *is minimized by matching the smallest value of* γ *with the largest value of* t, *the next larger value of* γ *with the next smaller value of* t, *and so on.*

Note that this property follows from the well-known result in linear algebra (Hardy et al., 1934) and leads to an $o(n\log n)$ algorithm (Panwalkar et al., 1982). It finds first the number of non-tardy jobs (k), and then calculates the positional penalties a, the optimal

sequence σ^* and the optimal due date d^*. The sequence generated by this algorithm is V-shaped. The optimal schedule starts at time $t^* = 0$ and has no inserted idle periods.

4.4 SINGLE-MACHINE

Consider the problem of scheduling n jobs on a single machine. A processing time p_j is associated with each job $j, j = 1,...,n$, and preemption is not allowed. Let $\sigma = ([1],[2],...,[n])$ be an arbitrary sequence of the jobs where $[j]$ is the j^{th} job in σ.

Optimal Solution Procedure

The single machine scheduling problem involves scheduling a set of tasks to a single resource. This is accomplished by determining a sequence that includes each task, and then assigning the tasks to the resource. Each task can be given a priority, ready time, processing time and due date. The value of the performance measures can be computed based on this information and the sequence of tasks.

The algorithm presented here consists of two phases. Phase 1 finds the number of nontrade job(k). Phase 2 calculates positional penalties Y, the optimal sequence, and the optimal due-date (d^*).

Phase 1

Step (i) Set $K' \leftarrow n(P_3 - P_1)/(P_2 + P_3)$.

Step (ii) Check if $K' > 0$. if Yes: go to step (iii)

If No: Set $d^* \leftarrow 0$. SPT sequence is optimal STOP.

Step (iii) Check K' is an integer

If Yes: set $K \leftarrow K'$.

Proceed to phase 2.

If No: set K equal to the smallest integer value $> K'$.

Phase 2

- **Step 2.1** Label position $j(1 \le j \le n)$ as

$$\gamma_j = \begin{cases} nP_1 + (j-1)P_2 & 1 \le j \le k \\ (n+1-j)P_3 & K+1 \le j \le n. \end{cases}$$

- **Step2.2.** Rank the positional labels γ_j in descending order of magnitude such that the largest γ_j is ranked 1 and the smallest γ_j is ranked n. Break ties arbitrarily.

- **Step 2.3.** Obtain the optimal sequence such that job i is scheduled in position j corresponding to γ_j ranked in position i.

- **Step 2.4** Set $d^* \leftarrow (t_{[1]} + t_{[2]} + ... + t_{[K]})$ STOP.

4.5 NUMERICAL EXAMPLE

Given Ten jobs with $t_1 = 2$, $t_2 = 5$, $t_3 = 8$, $t_4 = 11$, $t_5 = 14$, $t_6 = 16$, $t_7 = 19$, $t_8 = 22$, $t_9 = 23$, and $t_{10} = 25$. The penalties are $P_1 = 5, P_2 = 10$ and $P_3 = 15$.

$\Rightarrow$ From phase 1 the algorithms, we get $K' = 4$. The Ten positional labels and their ranks are as indicated below.

Position j	1	2	3	4	5	6	7	8	9	10
y_j	50	60	70	80	90	75	60	45	30	15
Rank i	7	5	4	2	1	3	6	8	9	10

Optimal sequence: 7, 5,4,2,1,3,6,8,9,10.

$$d^* = t_7 + t_5 + t_4 + t_2 = 49$$

One can easily verify that the total penalty is 6,755 units

4.6 PARALLEL-MACHINE

Consider the problem of scheduling n jobs on m parallel machines. Each job can be processed by any of the m machines taking into account the following constraints: a machine performs at

most one job at a time, and each job is performed at most by one machine at a time. If the machines are identical, they operate at the same speed, and processing time for job j is p_j. If the machines are uniform, each machine i has its own speed s_i, and processing time of job j on this machine will be $p_{ij} = p_j / s_i$. If the machines are unrelated, the speed of a machine is job-dependent, and processing time of job j on machine i is $p_{ij} = p_j / s_{ij}$. The goal is to determine an optimal due date and to schedule the jobs on the machines, so as to minimize an objective function. The type of objective function will be specified below for each of the problems under consideration. In what follows, it is assumed that preemption is not permitted and all jobs become available for processing simultaneously at time zero.

Parallel machine scheduling involves scheduling a set of tasks on two or more machines that work in parallel with each other. The machines perform identical operations and may or may not operate at the same pace. The basic analysis of the unrestricted version has been extended to models involving parallel machines. **Hall** [11], **Sundararaghavan** [12] examine the minimization of total absolute deviation with m identical machines operating in parallel. The multi-machine procedure assigns the m longest jobs to different machines.

Thereafter, the jobs are treated in non-increasing order of processing times and assigned $2m$ at a time among the machines. After all the jobs are assigned, Algorithm 1 is used to sequence the jobs on each machine. In addition, the four key properties apply to the optimal solution of the multi-machine model in the form

I. On each machine, there is no inserted idle time.

II. On each machine, the optimal schedule is V-shaped.

III. On each machine, one job completes at time d .

IV. The number of jobs assigned to each of the m machines is either $[n/m]$ *or* $[n/m]+1$

(Where $[x]$ denotes the integer portion of x). Let this number be denoted q . Then, on each machine, the b^{th} job in sequence completes at time d, where b the smallest integer is greater than or equal to $q/2$. The procedure to find the optimal due-date value for the parallel-machine problem can be derived in a similar fashion. Instead of differentiating (4.2.2). We take the first derivative of (4.2.1). The general penalty functions with respect to d , and equate the result to zero to obtain the first-order necessary condition for optimality given by (4.3.1). Thus the optimal due-date d^* should be set so that the numbers of non-tardy jobs are related to each other in accordance

with equation (4.3.1). For a given jobs sequence σ, let the jobs in σ be indexed according to non-decreasing completion time i.e. $C_1 \le C_2 ..... \le C_n$.

If the optimal due-date is set as $d^* = C_r$, then the necessary optimality condition equation (4.3.1) requires r to satisfy the following condition.

$$nP_1 + (r-1)P_2 - (n-r)P_3 = 0$$

$$\text{or } r = n(P_3 - P_1)/(P_2 + P_3) + P_2/(P_2 + P_3) \qquad (4.6.1)$$

Here we see that the optimal due-date should be set equal to the r^{th} completion time, where r is the least integer value not less than $n(P_3 - P_1)/(P_2 + P_3)$ for a given σ. It is clear, in this case, that $d^* = C_r$ is explicitly dependent on σ, whereas, in the single-machine case $d^* = C_{[r]}$ is a function of the job position r that is independent of σ. This is where the similarly between the single-machine and parallel machine problems stops and the difficulty of finding the point optimal due-date and job sequence starts.

4.7 HEURISTIC SOLUTION

A heuristic procedure to solve the parallel-machine problem therefore, we propose a simple heuristic based on the optimal results desired for the single machine problem. The heuristic is best described as an algorithm, as follows Heuristic algorithm.

- **Step 1.** Construct a list s of schedulable hobs in which the jobs are arranged in non-decreasing order of the processing times t_i. Initially, $|s| = n$.
- **Step 2.** Set $K = \lceil n/m \rceil$, where [x] is the smallest integer greater than or equal to x.
- **Step 3.** Set $r = [K(P_3 - P_1)/(P_2 + P_3)]$.
- **Step 4.** Set $\lambda_{j[i]} = KP_1 + (i-1)P_2$ for $1 \le i \le r$ and $1 \le j \le m$; $\lambda_{j[i]} = (K+1-i)P_3$ for $r+1 \le i \le k$ and $1 \le j \le m$.
- **Step 5.** Construct a list A of Position labels in which the labels are arranged in non-increasing order of $\lambda_{j[i]}$. Initially, $|A| = mk$.
- **Step 6.** Assign the first job in s to the position corresponding to the first label in A.
- **Step 7.** Delete the first hob and the first position label from S and A respectively. If $s \neq \phi$, go to Step 6.
- **Step 8.** A job sequence σ is generated. Calculate the completion time of the jobs in σ and re-index the jobs in non-decreasing order C_i i.e. $C_1 \le C_2 \le C_3 \le ..... \le C_n$. Set $d^* = C_r$, where $r = [n(P_3 - P_1)/(P_2 + P_3)]$.

- **Step 9.** Evaluate $f(d^*,\sigma)=\sum_{j\in M}\sum_{i\in N}\{P_1d^*+P_2E_{j[i]}+P_3T_{j[i]}$.

4.8 OPTIMAL SOLUTION

The computational requirement to obtain the true optimal solution by complete enumeration is prohibitive even for a small problem. It can easily be established that complete enumeration, in the worst case, requires $\sum_{n_1+n_2+...+n_m}(n_1!)(n_2!).....(n_m!)$ number of enumerations, where n_i denotes the number of jobs assigned to machine i, $1\le i\le m,$ and the summation is over all non-negative integer values of $n_1,n_2,.....,n_m$ such that $\sum_{1\le i\le m} n_i=n$.

Alternatively, the problem can be formulated as a mixed zero-one mathematical programmer as follows:

$$Min\ z=\sum_{1\le i\le n}\{P_1d+P_2E_i+P_3T_i\},$$

Subject to

$$C_i-d=T_i-E_i \qquad for\ 1\le i\le n$$

$$C_i=\sum_{1\le i\le m}\sum_{1\le k\le n}c_{jk}x_{ijk} \qquad for\ 1\le i\le n$$

$$C_{jk}=C_{j,k-1}+\sum_{1\le i\le n}t_ix_{ijk} \qquad for\ 1\le i\le m, 1\le k\le n$$

$$0\le\sum_{1\le i\le n}x_{ijk}\le 1 \qquad for\ 1\le i\le m, 1\le k\le n$$

$$\sum_{1 \le i \le m} \sum_{1 \le k \le n} x_{ijk} = 1 \qquad for\ 1 \le i \le n$$

$$x_{ijk} = 0 \ or\ 1 \qquad for\ 1 \le i \le n, 1 \le j \le m, 1 \le k \le n$$

$$C_{j0} = 0 \qquad for\ 1 \le i \le m,$$

where $x_{ijk} = 1(0)$ indicates job i is (not) assigned to position k of machine j; C_i is the completion time of job i; C_{jk} is the completion time of the job in position k of machine i.

4.9 CONCLUSION

We consider a total penalty for the n job, one machine scheduling problem in which all jobs have a common due date. The main objective is to determine the optimal value of this due date and an optimal sequence to minimize a total penalty function. Hence the problem belongs to class P. It is interesting to note that the determination of K, the number of non-tardy jobs, does not require the values of the processing times of the n jobs. We also show that the problem of scheduling n independent jobs on m parallel and identical machines. We further prove that the optimal due-date result can be generalized to the parallel-machines problem. We observe that the heuristic is very effective solution

4.10 REFERENCES

1. **Baker, K. R.:** Introduction to Sequencing and Scheduling. *John Wiley & Sons, New York,* 1974.

2. **Baker, K. R., and Bertrand, J. W. M.:** A Comparison of Due-Date Selection Rules, Industrial Engr. *Report 1980/02, University of Technology, Eindhoven, the Netherlands,* 1980.

3. **Conway, R. W., Maxwell, W. L., and Miller, L. W.:** Theory of Scheduling. *Addison-Wesley, Reading, Mass,* 1967.

4. **RinnooyKan, A. H. G., KAN.:** Machine Scheduling Problems: Classification, Complexity and Computations. *Martinus Nijhoff, the Hague,* 1976.

5. **French, S.:** Sequencing and Scheduling Ellis Harwood, *Chichester,* 1982.

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

7. **Eilon, S., and Chowdhury, I. J.:** Due-Dates in Job Shop Scheduling. *Int. J. Prod. Res.* 14, pp. 223-237, 1976.

8. **Jones, C. H.:** An Economic Evaluation of Job Shop Dispatching Rules. *Mgmt. Sci.* 20, pp. 293-307, 1973.

9. **Weeks, J. K.:** A Simulation Study of Predictable Due-Dates. *Mgmt. Sci.* 25, 363-373, 1979.

10. **Weeks, J. K., and Fryer, J. S.:** A Methodology for Assigning Minimum Cost Due-Dates. *Mgmt. Sci.* 23, pp. 872-881, 1977.

11. **Hall, N.:** Single and Multi-Processor Models for Minimizing Completion Time Variance. *Navai. Res. Logist. Quart.* 33, pp. 49-54, 1986.

12. **Sundararaghavan, P., and Ahed, M.:** Minimizing the Sum of Absolute Lateness in Single Machine and Multi-machine Scheduling. *Navai. Res. Logist. Quart.* 31, pp. 325-333, 1984.