Permutation Flowshop Scheduling with Earliness and Tardiness Penalties

Pankaj Chandra*, Peeyush Mehta**, Devanath Tirupati*

Working Paper No. 2004-07-06
July 2004

*Indian Institute of Management Ahmedabad, India
chandra@iimahd.ernet.in
devanath@iimahd.ernet.in
**Nanyang Technological University, Singapore
peeyush@ntu.edu.sg

The main objective of the working paper series of the IIMA is to help faculty members to test out their research findings at the pre-publication stage.

INDIAN INSTITUTE OF MANAGEMENT
AHMEDABAD-380 015
INDIA

Abstract

We address the permutation flowshop scheduling problem with earliness and tardiness penalties (E/T) and common due date of jobs. Large number of process and discrete parts industries follow flowshop type of production process. There are very few results reported for multi-machine E/T scheduling problems. We show that the problem can be sub-divided into three groups- one, where the due date is such that all jobs are necessarily tardy; the second, where the due date is such that it is not tight enough to act as a constraint on scheduling decision; and the third is a group of problems where the due date is in between the above two. We develop analytical results and heuristics for problems arising in each of these three classes. Computational results of the heuristics are reported. Most of the problems in this research are addressed for the first time in the literature. For problems with existing heuristics, the heuristic solution is found to perform better than the existing results.

Keywords: flowshop, earliness, tardiness, common due date

1. Introduction

In recent years, production managers have started laying emphasis on scheduling products as close as possible to their due dates. One of the driving reasons is the interest in Just-In-Time (JIT) manufacturing. The new interest in scheduling is to analyze the impact on the manufacturing costs of earliness, i.e., producing products before the due dates. One of the most obvious consequences of earliness is the cost incurred in finished goods inventory. Other reasons for reducing earliness would be limited storage space for finished goods, and the limited shelf life of products as in the case of chemicals and pharmaceuticals industries. Most of the existing scheduling literature has focused only on single performance measures such as lateness, tardiness, flowtime and number of tardy jobs etc. However, few have addressed multiple performance measures in the same objective function.

In this research, we consider the scheduling problem of minimizing earliness and tardiness (E/T) penalties in a flowshop type of production process. The motivation for this production environment is from our study of multi-stage production planning and scheduling problem (Chandra, Mehta and Tirupati, 2004), where the finished goods follow flowshop type of production process. Flowshop production environment exists in most of the process and discrete parts manufacturing industries. We consider common due dates of jobs. One of the reasons for this is to capture situations where large numbers of products are due from a single customer order with a common shipping date. The other reason is that in an assembly type of multi-stage production systems, intermediate products are prescribed common due date to avoid any downstream production delays. The notion of common due date is also consistent in production environment with high setup times where various customer orders of a product could be combined in a single production run and shipped on a common date.

The contribution of our research is new results in scheduling theory with earliness and tardiness penalties in a multi-machine production environment. We develop some analytical results and new heuristic algorithms to solve flowshop scheduling problems with E/T penalties. We also test the performance of the proposed heuristics and report their computational performance.

This paper is organized as follows. In the next section, we review the literature on scheduling with earliness and tardiness penalties. In section 3, we describe the scheduling problem addressed in this research. In section 4, we provide some existing results on the single machine E/T scheduling problem. We use some of these results in treatment of the flowshop E/T scheduling problem. In the following section, we develop the solution procedure for our scheduling problem. The results of the solution procedure are described in section 6. Finally, we summarize the paper in the last section.

2. Literature Review

The study of earliness and tardiness penalties is a relatively new area of research in scheduling theory. The variety in E/T scheduling literature is generated from the assumptions made about due dates and penalty costs. However, most of the results in E/T scheduling are for single machine problems only. There is very limited research reported on multi-machine production environment with E/T penalties.

The issue which stands out in E/T scheduling research is that how the scheduling decisions are constrained by the due dates. Considering that due dates are common for all jobs, problems which have due date late enough so as not to influence the scheduling decisions are called unrestricted due date problems. If the due date constrains the scheduling decisions, then it is referred as the restricted due date problem. Kanet (1981) provided the first set of results that defined unrestricted common due date in scheduling with E/T penalties. The objective in this paper was to minimize absolute deviation of job completion times from the due date. Kanet provides an algorithm to determine an optimal solution solvable in polynomial time. The optimality conditions and alternate optimal solutions of single machine are also discussed in Sundararaghavan and Ahmed (1984), Hall (1986), and Bagchi, Chang and Sullivan (1986). The analysis of restricted version of the problem is due to Bagchi, Chang and Sullivan (1986). NP-completeness of the restricted due date single machine problem was proved by Hall, Kubiak and Sethi (1991). The single machine E/T problem has also been investigated with objectives like weighted penalties, non-linear penalties, completion time variance etc. A comprehensive review of the problem can be found in Baker and Scudder (1990).

An issue that is beneficial in scheduling problems with earliness penalties is that of inserted idle time (IIT). Most of the E/T work in scheduling does not consider IIT either by restricting the solution to be a non-delay schedule or by assuming a common due date for all jobs. For the $n|1|d_i=d|\Sigma(E_i+T_i)$ (i.e, common due date problem), Cheng and Kahlbacher (1991) proved that it is unnecessary to consider schedules with inserted idle time except prior to the first job in the schedule. Kanet and Sridharan (2000) provide a review of IIT scheduling. However, they do not consider the review of Baker and Scudder (1990), as these papers are restricted to non-IIT and non-delay schedules. Both the review papers, Kanet and Sridharan, and Baker and Scudder observe that the essence of E/T problem lies in its non-regular performance measure. Imposing the restriction of no inserted idle time diminishes the objective.

In the multi-machine production environment, Koulamas (1994) has shown NP-hardness of $F|\,|\,\Sigma T_i$ problem for $m \geq 3$. The above complexity result coupled with the nature of flowshop has limited the possibility of developing efficient solution algorithms for $F|\,|\,\Sigma T_i$. Since $F|\,|\,\Sigma T_i$ is NP- Hard, $F|\,|\,\Sigma(E_i+T_i)$ is also NP-Hard. Research on E/T penalties in multi-machine setting is very scanty. Gowrishankar et al. (2001) considered minimizing the completion time variance and the sum of squares of completion time deviations from a common due date. They develop lower bound for both the problems. Using the lower bound, they propose branch and bound algorithms for the two problems. For larger problems, they propose heuristics for both types of problems.

In a multi-machine production environment, there is no work reported in the literature that investigates the minimization of absolute deviation of job completion times from the common due date. In the next section, we describe the scheduling problem addressed in this research.

3. Scheduling Problem

In a multi-machine production environment, the E/T costs in scheduling are function of the schedule of jobs on the last machine. Considering that m is the last machine in a flowshop, tardiness of a job T_i is defined as: $T_i = max\ (C_{im} - d,\ 0)$, where C_{im} is the completion time of job i on machine m and d is the common due date of the job. Earliness of a job E_i is defined as: E_i = max ($d - C_{im}$, 0). The scheduling problem in this research is to determine a sequence of all jobs and their schedule with minimum earliness and tardiness costs. The schedule of a job comprises determination of S_{ij}, the start time of job i on machine j, and C_{ij}. The objective is to minimize $\sum(E_i + T_i)$ $\forall i = 1$ to n.

We described in section 1 the reason for considering common due date for all jobs. In order to derive scheduling decisions, we consider permutation sequence of jobs on the machines. The motivation for this is that usually in process industries, the desired production quantity of a product is achieved with production runs of small batches of known process yields. Setup times are usually very high during product changeover, and only minor setup is incurred when a new batch of same product is produced. A batch has its own identity and a specific schedule. This essentially means that a product schedule comprises schedule of its each batch in a production run. Since each machine in a flowshop would have same sequence of batches of a product, it is appropriate to consider permutation flowshop in the scheduling problem and treat each batch of a product as a job. Next, we provide a mixed integer-programming model that addresses the scheduling decisions.

3.1 Scheduling Problem Formulation

Indices and index sets

i	=	index of jobs
j	=	index of machines
N	=	set of jobs, $\{i \mid i=1,2,\ldots,n\}$
S	=	set of machines, $\{j \mid j=1,2,\ldots,m\}$

Parameters

d	=	common due date of jobs
p_{ij}	=	processing time of job i on machine j

Variables

S_{ij}	=	start time of job i on machine j
C_{ij}	=	completion time of job i on machine j
T_i	=	tardiness of job i, $T_i = max(C_{im} - d, 0)$
E_i	=	earliness of job i, $E_i = max(d - C_{im}, 0)$
y_{ik}	=	$\begin{cases} 1, \text{ if job } i \text{ is before job } k \text{ in a sequence, } i, k \in N \\ 0, \text{ otherwise} \end{cases}$

The scheduling problem can be formulated as follows:

$$\min\ Z = \sum_i E_i + T_i = \sum_i \left| C_{im} - d \right|$$

subject to:

$$C_{ij} \geq C_{ij-1} + p_{ij} \qquad \forall i \in N, j \in S \qquad (1)$$

$$S_{kj} - (S_{ij} + p_{ij}) + M(1 - y_{ik}) \geq 0 \qquad \forall i, k \in N, j \in S \qquad (2)$$

$$S_{ij} - (S_{kj} + p_{kj}) + M\ y_{ik} \geq 0 \qquad \forall i, k \in N, j \in S \qquad (3)$$

$$C_{im} - d = T_i - E_i \qquad \forall i \in N \qquad (4)$$

$$C_{ij} = S_{ij} + p_{ij} \qquad \forall i \in N, j \in S \qquad (5)$$

$$C_{ij}, S_{ij}, E_i, T_i \geq 0 \qquad (6)$$

$$y_{ik} \in \{0,1\}$$

Constraint 1 is the operation precedence constraint for a job. It ensures that an operation cannot start until the previous operation has been completed. Constraints 2 and 3 indicate job precedence at a machine. They ensure that if a job *i* is scheduled before job *k*, then at each machine job *k* is started only after job *i* is completed. Constraint 4 determines E_i or T_i of a job, as the case may be. Constraint 5 indicates that preemption is not allowed for a job and determines the start times of each job at each machine.

In subsequent sections we describe the solution procedure to solve the scheduling problem. We begin with discussing some results for a single machine E/T common due date problem in the next section. These results form the basis of developing solution procedure for the flowshop E/T common due date problem.

4. Existing Single Machine Results

In this section, we revisit from literature results on single machine scheduling problem of minimizing absolute deviation of job completion times from their common due date. In the next section, we extend some of these results to obtain analytical results for the flowshop E/T scheduling problem. The detailed description of results on single machine E/T scheduling problem is also available in Baker and Scudder (1990).

Let the unrestricted due date for single machine (discussed in section 2) be d_0, and let *SUD(d)* be the single machine E/T problem for common due date, $d \geq d_0$. Let us recall that the unrestricted due date makes the single machine problem unconstrained, i.e., the due date is not early enough to act as a constraint on the scheduling decision. Also, the optimal solution to *SUD(d)* is available in polynomial time. If p_i the processing time of job i and jobs are arranged such that $p_1 \leq p_2 \leq p_3 \ldots \leq p_n$, the E/T single machine problem is unrestricted, if due date d is such that:

$d \geq d_0 = p_2 + p_4 + p_6 + \ldots\ldots + p_{n-4} + p_{n-2}, + p_n,$ *if n is even.*

$d \geq d_0 = p_1 + p_3 + p_5 + \ldots\ldots + p_{n-5} + p_{n-3} + p_n,$ *if n is odd.*

The optimal sequence for *SUD(d)* is:

(n, n-2, n-4,.....,2,..,1,....3,.......n-3, n-1), if n is even.

(n, n-2, n-4,.....,1,..,2,.... 4,.......n-3, n-1), if n is odd.

Under these conditions, the optimal solution of *SUD(d)* has following properties (Baker and Scudder, 1990):

1. There is no idle time in the schedule. This means that if job j immediately follows job i in the schedule with completion time, $C_j = C_i + p_j$
2. The optimal schedule is V Shaped. Jobs for which $C_i \leq d_0$ are sequenced in non-increasing order of processing time, while jobs for which $C_i > d_0$ are sequenced in non-decreasing order of processing times. Raghavachari (1986) establish the V-shape of an optimal schedule for any common due date.
3. One job completes precisely at the due date, i.e., $C_i = d_0$ for some i.

Let the optimal sequence of *SUD(d)* be *1,2,...e-1, e, e+1,....n.* In this sequence, e is the job that finishes at common due date d, i.e., $C_e = d$ and $S_e = C_e - p_e$, where C_e, S_e are the completion time and start time of job e respectively. As there is no idle time in this schedule, $C_{e-1} = S_e$ and $S_{e-1} = C_{e-1} - p_{e-1}$. The schedule of the optimal sequence is determined in this manner.

We would like to state here that there could be alternate optimal sequences of *SUD(d)* for any $d \geq d_0$, although the optimal value of *SUD(d)* remains same. The optimal sequence shown above is assumed to be at $d = d_0$. It is difficult to obtain all alternate optimal sequences for $d > d_0$. However, all the alternate optimal sequences can be obtained for

$SUD(d)$ at $d = d_0$. It is to be noted that there will be alternate optimal sequences at $d = d_0$, only if, the processing times of any two jobs are same. The set of all alternate optimal sequences at $d = d_0$ is used later in solving the flowshop E/T scheduling problem. The procedure to generate all alternate optimal sequences at $d = d_0$ (GAOS) is described in Appendix 1.

Similarly there are results for single machine E/T problem for restricted due date, i.e., $d < d_0$ (Hall, Kubiak and Sethi, 1991). The restricted due date is so early that it influences the scheduling decisions. Thus, the treatment of single machine E/T problem is guided by the constraint that distinguishes the restricted and unrestricted problems. In the next section, we derive the constraints that classify the scheduling problem as restricted and unrestricted in a flowshop setting. Subsequently we outline the solution procedure to solve the flowshop E/T problem.

5. Solution Procedure for Flowshop E/T Problem

In this section, we discuss the procedure for solving the flowshop E/T scheduling problem considered in this research. In order to do this, we use the treatment of single machine E/T problem in literature as the building block to solve the flowshop E/T problem. First, we will derive the constraints that make the flowshop problem restricted or unrestricted. Then, we categorize the flowshop E/T problem into three Sub-Problems and develop solution procedure for each of the Sub-Problem. We begin with deriving the unrestricted due date.

Notation

S	=	index of sequences of jobs, $s = 1,2,\ldots l$
$S(m, d_0)$	=	set of optimal sequences of $SUD(d_0)$ at last machine m with common due date d_0. The set is generated by procedure described in Appendix 1.
$E(s, d_0)$	=	set of early and on-time jobs in sequence s with common due date d_0, $s \in S(m, d_0)$.
$T(s, d_0)$	=	set of tardy jobs in sequence s with common due date d_0, $s \in S(m, d_0)$.
$r(s, d_0)$	=	schedule of optimal sequence s, consisting of S_i and C_i $\forall i$,

		$s \in S\ (m,d_0)$. Schedule is generated as described in the procedure above in this section.
$Z_1\{r\ (s,\ d_0)\}$	=	earliness and tardiness costs of schedule $r\ (s,\ d_0)$.
$F(s)$	=	flowshop schedule of sequence $s,\ s\ \in S(m,d_0)$. $F(s)$ is determined as follows. Let the sequence be $1,2,\ldots\ldots n$. $S_{11} = 0$, for $i = 1\ to\ n$ for $j = 1\ to\ m$, $S_{ij} = max\ \{C_{ij-1},\ C_{i-1j}\}$ $C_{ij} = S_{ij} + p_{ij}$
$M_{F(s)}$	=	Makespan of schedule $F(s)$; $M_{F(s)} = C_{nm},\ s\ \in S\ (m,\ d_0)$. This is the completion time of last job in the sequence.

We define k as the sequence with minimum makespan, i.e., $k = \arg \min\limits_{s \in S(m,d_0)} M_{F(s)}$. The unrestricted due date d_1 in permutation flowshop environment is then defined as $d_1 = M_{F(k)} - \sum\limits_{j \in T(k,d_0)} p_{jm}$. The second term is the sum of tardy jobs in sequence k. This essentially means that for a common due date $d \geq d_0$, the flowshop E/T problem is unconstrained and the due date does not influences the scheduling decisions.

Next, we develop the restricted due date d_2 in a permutation flowshop setting. Let us define $a = \arg \min\limits_{i} \sum\limits_{j=1}^{m} p_{ij} \quad \forall\ i = 1,2,\ldots.n$, where a is the minimum of sum of processing times of job at all machines amongst all jobs. We call this sum as the restricted due date, i.e., $d_2 = \sum\limits_{j=1}^{m} p_{aj}$. By definition, no job can be early with due date $d \leq d_2$. The above discussion gives rise to another range of due date, that is in between the restricted and unrestricted due date, and we call it as the intermediate due date. Thus, for flowshop E/T problem with common due date, we define these Sub-Problems for $d \geq d_1$(unrestricted due date), $d_2 < d < d_1$ (intermediate due date) and $d \leq d_2$ (restricted due date). On the basis of this classification of due dates, we have decomposed the flowshop E/T problem into three Sub-Problems as shown in Figure1.

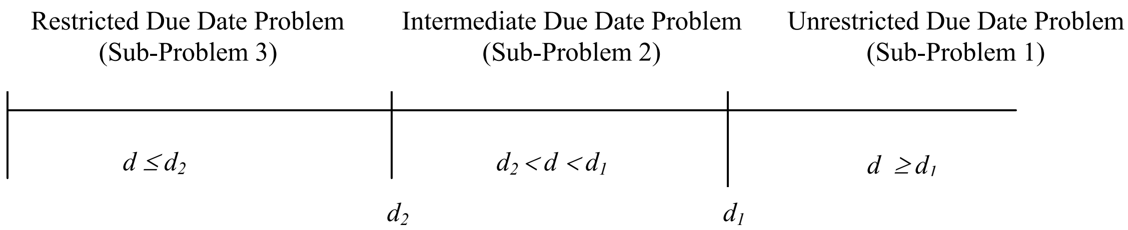

Figure 1: Flowshop E/T Problem Decomposition Based on Due Dates

Sub-Problem 1 is the flowshop E/T problem defined over the unrestricted common due date $d \ge d_1$. Sub-Problem 2 is flowshop E/T problem defined over the intermediate due date, $d_2 < d < d_1$ and Sub-Problem 3 is the flowshop E/T problem defined over the restricted due date $d \le d_2$. As discussed earlier, Sub-Problem 3 has a special structure by definition of d_2 that all jobs will be necessarily tardy. In the following sub-sections, we describe each of the Sub-Problems and develop the solution procedures. In sub-section 5.1, which follows next, we solve Sub-Problem 1.

5.1 Sub-Problem 1: Flowshop E/T Problem for Unrestricted Common Due Date

In this sub-section, we develop the solution procedure for solving the permutation flowshop E/T problem for unrestricted common due date, $d \ge d_1$. The problem is to determine a flowshop schedule with minimum E/T costs. The objective of Sub-Problem 1 is to minimize E/T penalties, i.e., $Minimize\ Z = \sum_i E_i + T_i = \sum_i \left| C_{im} - d \right|$, where C_{im} is the completion time of job i on the last machine m.

One of the optimal properties of *SUD(d)* is that there is no idle time in the schedule. If there is any idle time, it should be removed while maintaining the feasibility of the schedule. The procedure to *remove idle time* (RIT) in the schedule *F(s)* at the last machine is described in Appendix 2. This procedure will be used in the development of solution procedure for solving Sub-Problem 1. We now state a theorem to determine optimal solution for Sub-Problem 1.

Theorem 1: *For a flowshop E/T problem with common due date $d \geq d_1$, there is an optimal sequence k with $Z\{F(k)\} = Z_1\{r(k,d_0)\}$.*

Proof: By definition of *SUD(d_0),* sequence k is optimal for $d \geq d_0$. It follows that for $d \geq d_0$, $Z_1\{r(k, d)\} = Z_1\{r(k,d_0)\}$. By definition, $d_1 \geq d_0$. Thus for $d \geq d_1$, sequence k is optimal for *SUD(d)* and $Z_1\{r(k,d_1)\} = Z_1\{r(k,d_0)\}$. $Z\{F(k)\}$ is function of completion time of jobs at machine *m*, i.e., $Z\{F(k)\} = \sum_{j=1}^{n} |C_{jm} - d_1|$ for $d = d_1$. It follows that $Z\{F(k)\} \geq Z_1\{r(k,d_1)\}$ as $Z_1\{r(k,d_1)\}$ is optimal for $d = d_1$.

In schedule *F(k)* at machine *m*, if $S_{im} = C_{i-1.m}$ $\forall$ $i = n, n-1, n-2, \ldots, 2$, sequence *k* has all optimal properties of *SUD(d) at* $d = d_1$. If $S_{i.m} \geq C_{i-1.m}$ $\forall i = n, n-1, n-2, \ldots, 2,$ this idle time can be removed by the procedure RIT developed in Appendix 2.

It follows that sequence *k* has now all properties of *SUD(d)* at $d = d_1$. Thus, $Z\{F(k)\} = Z_1\{r(k, d)\}$ at $d = d_1$. If d_1 is increased to $d_1 + \Delta$, the optimal schedule at stage *m* would be $C_{im} = C_{im} + \Delta$ for $i = n\text{-}1$ to *1* and $C_{nm} = M_{F(k)} + \Delta$. For $d > d_1$, all properties of *SUD(d)* hold. Hence for $d \geq d_1$, $Z\{F(k)\} = Z_1\{r(k, d_0)\}$ and sequence *k* is optimal.

Q.E.D.

The theorem given above provides the optimal solution for Sub-Problem 1. We would like to state that the value of unrestricted due date d_1 in Sub-Problem 1 is determined on the basis of set of all optimal sequences of single machine E/T problem at $d = d_0$. As mentioned earlier, it is difficult to obtain optimal sequences for single machine E/T problem for $d > d_0$. In that sense the value of d_1 could be made tighter. This is because some of the optimal sequences for $d > d_0$ could have lesser makespan than $M_{F(k)}$, and d_1 is a function of $M_{F(k)}$. In the next sub section, we describe Sub-Problem 2 and develop its solution procedure.

5.2 Sub-Problem 2: Flowshop E/T Problem for Intermediate Common Due Date

In this sub-section, we provide the heuristic algorithm for solving Sub-Problem 2. The objective of Sub-Problem 2 is same as that of Sub-Problem 1, i.e., $Minimize\, Z = \sum_i E_i + T_i = \sum_i |C_{im} - d|$. The difference between Sub-Problems 1 and 2 is in the value of the common due date *d*. The common due date value for Sub-Problem 2 is between

d_2 and d_1, i.e., $d_2 < d < d_1$. Garey et al. (1976) provide proof of NP-completeness of this problem. Next, we describe the proposed heuristic algorithm to solve Sub-Problem 2.

5.2.1 Heuristic Algorithm (H1) for Sub-Problem 2

The heuristic for solving Sub-Problem 2 is based on deriving a permutation sequence of jobs at the bottleneck machine. Bottleneck machine is identified as the machine that requires maximum sum of processing time of all jobs amongst all machines. We solve the single machine E/T problem at the bottleneck machine. The pre-bottleneck processing times of a job is captured by considering release dates of job at the bottleneck machine. The release date of a job in this problem is defined as the earliest time at which the job is available for processing at the bottleneck machine. The post-bottleneck processing times of a job is captured by determining the due date of a job at the bottleneck. The resulting problem is single machine E/T problem with release dates and distinct due dates, $n/1/r_i/\Sigma(E_i+T_i)$. We solve this single machine problem at the bottleneck machine. To solve this, we use results on $n/1/r_i/\Sigma(E_i+T_i)$ by Chu (1992) and Chu and Portmann (1992). They derive a sequence of jobs on single machine. In our heuristic, using a priority function (defined in the detailed heuristic steps), a job is selected and appended to a partial sequence. Schedule of the partial flowshop sequence is developed subsequently. Based on this schedule, release dates and due dates of a job are updated at each iteration of appending the job. The schedule of the complete permutation sequence is then modified to improve earliness and tardiness costs. In the end, local neighborhood search procedure (tabu search) is applied to improve the solution. The detailed steps of the heuristic for solving Sub-Problem 2 (H1) are provided in Appendix 3. Next, we describe the solution procedure for solving Sub-Problem 3.

5.3 Sub-Problem 3: Flowshop Tardiness Problem for Common Due Date

We now discuss the Sub-Problem 3 of minimizing earliness and tardiness penalties in a flowshop for common due date, $d \leq d_2$. This Sub-Problem has a special structure, by definition of d_2, no job is early. Thus, the problem reduces to one of minimizing tardiness. Since the due date in our problem is common for all jobs, minimizing tardiness is same as minimizing flowtime, if all jobs are necessarily tardy. Further, since all jobs are simultaneously available, the minimizing flowtime problem is same as minimizing completion time. Thus our problem is to minimize tardiness or flowtime or completion time of all jobs. We now derive the analytical solution for Sub-Problem 3. We begin by defining some new terms.

Notation

q	=	index of sequences of jobs
S	=	set of permutation flowshop sequences
d, d'	=	common due date of jobs
$\sigma(q, d)$	=	permutation flow shop schedule of sequence q and due date d, $q \in S$.
$Z\{\sigma(q, d)\}$	=	Early/Tardy cost of schedule $\sigma(q, d)$,

$$Z\{\sigma(q,d)\} = \sum_{j=1}^{n} \left| C_{im} - d \right|$$

$$k = \arg\min_{i} \sum_{j=1}^{m} p_{ij}$$

$$d_2 = \sum_{j=1}^{m} p_{kj}$$

Proposition 1: *In a flowshop E/T problem with common due date d, an optimal sequence s for $d = d_2$ is optimal for $d < d_2$.*

Proof: Suppose the optimal sequence s for $d = d_2$ is not optimal for $d < d_2$. From definition of d_2, in any flowshop sequence q, no job is early ($E_i = 0$, $\forall\, i = 1,2,\ldots,n$) for $d = d_2$. Hence schedule $\sigma(q, d)$ has regular performance measure (non-decreasing in C_{ij}) for $d = d_2$. For regular performance measures, the cost of any schedule with inserted idle time $t = \Delta$ can be improved by removing Δ as C_{ij} $\forall\, i, j$ are reduced by $t = \Delta$. Hence we consider $\sigma(q, d_2)$ without inserted idle time and all jobs are scheduled as early as possible. $\sigma(q, d_2)$ is derived as follows:

for $i = 1$ to n

  for $j = 1$ to m

$S_{11} = 0$

$S_{ij} = \max\{C_{ij-1}, C_{i-1j}\}$

$C_{ij} = S_{ij} + p_{ij}$

$$Z\{\sigma(q, d_2)\} = \sum_{j=1}^{n} \left| C_{im} - d_2 \right|$$

From definition of $Z\{\sigma(q, d_2)\}$, it can be seen that:

for $d = d_2\text{-}1$, $Z\{\sigma(q, d)\}$ increases by n,

for $d = d_2\text{-}2$, $Z\{\sigma(q, d)\}$ increases by $2n$,

for $d = d_2\text{-}x$, $Z\{\sigma(q, d)\}$ increases by xn.

Thus, for any $d < d_2$, $Z\{\sigma(q, d)\}$ increases by $(d2\text{-}d)n$,

Hence, for $d < d_2$, $Z\{\sigma(q, d)\} = Z\{\sigma(q, d_2)\} + (d_2\text{-}d)n$

Now consider an optimal sequence s for $d = d_2$. Suppose s is not optimal for a due date d' where $d' < d_2$. Consider another sequence $s1$, which is optimal for $d' < d_2$. Then we have,

$$Z\{\sigma(s, d')\} = Z\{\sigma(s, d_2)\} + (d_2\text{-}d')\, n \qquad (1)$$

$$Z\{\sigma(s1, d')\} = Z\{\sigma(s1, d_2)\} + (d_2\text{-}d')\, n \qquad (2)$$

If s is not optimal for d',

$$Z\{\sigma(s, d')\} > Z\{\sigma(s1, d')\} \qquad (3)$$

From (1), (2) and (3) ,

$$Z\{\sigma(s, d_2)\} + (d_2\text{-}d')n > Z\{\sigma(s1, d_2)\} + (d_2\text{-}d')n$$

Thus, $Z\{\sigma(s, d_2)\} > Z\{\sigma(s1, d_2)\}$. This is a contradiction as s is an optimal sequence for $d = d_2$. Hence s is an optimal sequence for $d < d_2$.

This result has implications that the optimal solution of flowshop tardiness problem for common due date, $d \leq d_2$ (Sub-Problem 3) remains the same. We develop a heuristic for solving this problem. Several researchers have investigated the problem of minimizing tardiness, flowtime, and completion time in permutation flowshops (Nawaz et al., 1983; Rajendran, 1993; Woo and Yim, 1998). The equivalence of these three objectives for Sub-Problem 3 was discussed above.

The concept behind the heuristic is the same as used in heuristic algorithm for Sub-Problem 2. We derive a permutation flowshop sequence at the bottleneck machine. The one minor difference between the heuristics for Sub-Problems 2 and 3 is that the priority function for a job is determined differently. This is because in Sub-Problem 3 we are minimizing only tardiness, whereas E/T costs are minimized in Sub-Problem 2. Secondly, the steps for improving earliness and tardiness costs of heuristic of Sub-Problem 2 are not required. The steps of the heuristic solution of Sub-Problem 3 (H2) are explained in Appendix 4. In the next section, we discuss the computational results obtained by using these solution procedures.

6. Computational Results

As discussed in section 5, we have analytically derived optimal solution for Sub-Problem 1. In this section, we discuss the computational results on Sub-Problems 2 and 3. We will describe the lower bound on these Sub-Problems, the experiment design and the computational performance of the heuristics developed for Sub-Problems 2 and 3.

6.1 Lower Bound on Sub-Problem 2

$O_j(i)$	=	sum of i shortest processing times on machine j amongst all Jobs
LBC_i	=	lower bound on the completion time of job i on machine m.
C_{im}	=	completion time of job i on machine m
$LBET_i$	=	lower bound on earliness and tardiness of job i

In a permutation flowshop, the completion time of the i^{th} job on the last machine m, *i.e.,* for any sequence, $LBC_i \geq \max_{1\leq j\leq m}\left\{O_j(i) + \min_i \sum_{l=1}^{m} p_{il} - \min_i p_{ij}\right\}$. $O_j(i)$ is a lower bound on the time needed to process i jobs on machine j. Therefore, C_{im} is not less than the sum of $O_j(i)$ and the minimum processing times among all jobs on machine 1 through m except machine j. Since this is true for all machines, the LBC_i is a valid lower bound on completion time of i^{th} job on last machine of any sequence. LBC_i is provided by Kim (1995). The lower bound on earliness and tardiness of job i is given by:

$LBET_i = \max\{d - LBC_i, 0\} + \max\{LBC_i - d, 0\}$. The first sum is the lower bound on earliness, and the second sum is lower bound on tardiness. It is difficult to determine the lower bound on earliness. Hence, we consider $LBET_i = \max\{LBCi - d, 0\}$. Next, we describe the experiment design for measuring the computational performance of Sub-Problem 2.

6.2 Experiment Design for Sub-Problem 2

The procedures described in the heuristic algorithm for solving Sub-Problem 2 are applied to benchmark problems in the literature on flowshop scheduling (Taillard, 1993). The parameters used in the experiments are shown in the Table 2 below.

Number of jobs, *n*	*n = 5, 10, 20, 50, 80, 100*
Number of machines, *m*	*m = 5, 10, 15, 20*
Number of instances, *I,* of test problems	*I = 50*
Processing time of a job on a machine in each instance, p_{ij}	Random number uniformly distribution between 1 and 99.
Number of tabu iterations	50, 60, 70, 80
Tabu tenure	Random number between 5 and 10

Table 2: Parameters in Experiment Design of Sub-Problem 2

For small problems, optimal solution is obtained using Branch and Bound algorithm. The MIP model is developed in GAMS with CPLEX solver. The performance of the heuristic for small problems is compared with optimal solution. For large problems, the heuristic solution is compared with the lower bound. The performance measure of the heuristic is the average percentage deviation of the heuristic solution from the optimal solution in small problems P_{HO}, and from the lower bound in large problems P_{HL}.

We define,

Z_{HI}: Objective value of heuristic solution of instance *I*

Z_{OI}: Objective value of optimal solution of instance *I*

Z_{LBI}: Lower bound of the instance *I*

For smaller problems (*n =5*, *10*; *m =5*)

$$P_{HO} = \frac{1}{I}\left(\sum_{I} \frac{Z_{HI} - Z_{OI}}{Z_{OI}}\right)100$$

For large problems (*n > 10*)

$$P_{HL} = \frac{1}{I}\left(\sum_{I} \frac{Z_{HI} - Z_{LBI}}{Z_{LBI}}\right)100$$

LBC_i is a weak lower bound (Kim, 1995). It is difficult to estimate the lower bound on earliness. Thus, $LBET_i$ is a very weak lower bound on earliness and tardiness. This is verified for small problems (*n = 5*, *10*; *m =5, I = 50*). The average percentage deviation of optimal solution from the lower bound is 326 percent for 5- jobs, and 284 percent for 10-

jobs. However, for ($n = 5, 10; m = 5, I = 50$), P_{HO} is 0.894 percent and 1.126 percent for 5-jobs and 10-jobs respectively. The common due date considered for this analysis is $d = (d_1+d_2)/2$. The observations are encouraging for measuring heuristic performance, as the optimal solution also has large deviation from the lower bound.

The performance of the heuristic for smaller problems is also compared with optimal solution with a random common due date between d_1 and d_2. This is done to evaluate the quality of heuristic solution in the entire range of intermediate due date. For ($n = 5, 10; m = 5, I = 50$), P_{HO} is 0.846 percent and 1.247 percent for 5-jobs and 10-jobs respectively.

Since the lower bound of Sub-Problem 2 is very weak, the performance measure of the heuristic for larger problems is tested for common due date value d_1(obtained in Sub-Problem 1). This is because we have optimal solution of flowshop E/T problem for common due date d_1, obtainable in polynomial time. The results of this comparison are indicated in Table 3. The results in Table 3 indicate the average percentage deviation of optimal solution at $d = d_1$ (obtained from analytical solution for Sub-Problem 1) from the heuristic solution. The results of Table 3 indicate that the performance of heuristic H1 is good, as the maximum average percent deviation of the optimal solution from lower bound is found to be 1.744 percent.

	Machines			
Jobs	**5**	**10**	**15**	**20**
5	0.000	0.000	0.235	0.000
10	0.084	0.081	0.099	0.276
20	0.074	0.020	0.012	0.023
50	0.323	0.153	0.152	0.146
80	0.865	0.642	0.617	0.644
100	1.744	1.168	1.175	1.129

Table 3:Average Percentage Deviation of Optimal Solution from Heuristic Solution for Common Due Date, d = d_1.

When the number of tabu iterations is increased, the results improve. The average percentage deviation is found to be reducing. This, however, increases the computational time to solve the problem. The improvement in results with increase in number of tabu iterations is shown in Figure 2 for ($n = 50, m = 5, I = 50$). As seen in Figure 2, the solution at 100 tabu iterations is around 70 percent better than the solution at 50 tabu iterations. In the next sub-section, we discuss the results of Sub-Problem 3.

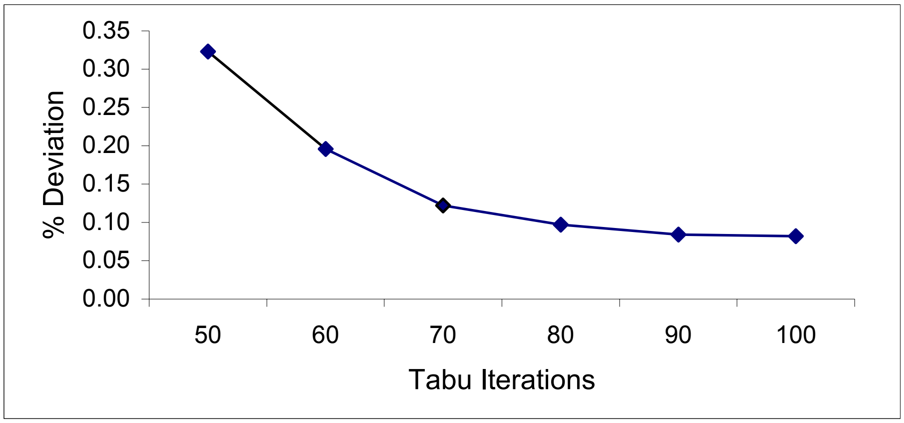

Figure 2: Improvement in the solution with Increase in Number of Tabu Iterations

6.3 Results of Sub-Problem 3

In this section, we discuss the results of flowshop E/T problem with restricted common due date, i.e., $d < d_2$. The special structure of Sub-Problem 3 was discussed in Section 4. Sub-Problem 3 determines a permutation flowshop schedule of all jobs with minimum tardiness costs. The objective of this problem is to minimize tardiness of jobs. Because of the common due date and the property that no job is early, the objective of the problem is same as that of minimizing flowtime and minimizing completion time. As a result, we can use one of the better-known lower bounds in literature, of flowshop completion time problem, as the lower bound of Sub-Problem 3. The best-known lower bound of flowshop completion time problem is due to Ahmadi and Bagchi (1990). Let the value of this lower bound be called Z_{LBAB}.

There are several results in the literature on flowshop problems with an objective of minimizing tardiness, flowtime or completion time of jobs. Due to the equivalence of these objectives in the case of Sub-Problem 3, we compare some of the existing best results to valuate the performance of our heuristic (H2) for solving Sub-Problem 3. We consider following three heuristics existing in the literature:

1.	NEH	Nawaz et al. (1983)
2.	RZ	Rajendran and Ziegler (1997)

3. WY Woo and Yim (1998)

We determine average percentage deviation from lower bound LBAB on each of the three heuristics (NEH, RZ, and WY). On the same instances we test our heuristic (H2), which was described in Section 5. We also propose two more heuristics by applying tabu search procedure on heuristics RZ and WY. These heuristics are RZT and WYT. Table 4 indicates the performance of existing heuristics and the proposed heuristics for various jobs

	RZ	NEH	WY	RZT	WYT	H2
Jobs	Machines	5				
5	7.212	7.964	7.283	6.922	6.922	6.922
10	10.999	13.359	11.865	10.246	10.351	10.402
20	16.091	19.516	17.090	14.511	14.975	15.306
50	21.267	27.178	21.792	19.554	19.993	20.705
80	22.547	30.702	22.351	20.633	20.840	21.881
100	23.350	31.531	23.078	21.559	21.588	22.976
Jobs	Machines	10				
5	8.385	9.278	8.685	8.247	8.247	7.892
10	14.729	16.082	15.335	14.041	14.091	13.530
20	21.389	23.438	22.371	19.709	20.264	19.728
50	29.126	32.444	29.842	27.178	27.799	27.770
80	30.706	35.337	30.918	28.799	29.343	29.630
100	32.242	37.024	32.353	30.160	30.871	30.731
Jobs	Machines	15				
5	8.397	10.423	9.001	8.308	8.331	8.073
10	14.698	16.544	15.389	14.015	14.122	13.438
20	22.678	25.279	23.527	21.411	21.324	20.824
50	32.412	34.256	33.124	30.642	30.936	30.314
80	35.110	36.258	35.620	33.459	33.988	33.584
100	37.735	38.451	37.808	35.993	36.312	36.362
Jobs	Machines	20				
5	7.856	9.635	8.203	7.764	7.764	7.432
10	14.171	15.396	14.706	13.420	13.475	13.086
20	22.880	24.587	23.714	21.858	22.073	21.653
50	33.289	35.213	34.207	31.820	32.346	31.847
80	37.548	39.254	37.957	35.987	36.315	36.524
100	39.399	40.267	39.406	38.413	38.039	37.883

Table 4: Average Percentage Deviation of Heuristic Solution from Lower Bound (P_{HL})

and machine combinations. The parameters of experiment design to measure the heuristic performance are same as used in the experiment design for Sub-Problem 2.

Table 4 indicates the comparison of three existing heuristics and three new heuristics developed to solve Sub-Problem 3(i.e., heuristics NEH, RZ, WY, RZT, WYT and H2). As seen in Table 4, where $Z_{LBI} = Z_{LBAB}$ for each instance I, P_{HL} in all the cases is better for proposed heuristics as compared to the existing heuristics In all the heuristics, the average percentage deviation from lower bound increases with the number of jobs.

7. Conclusions

In this research, we have solved the permutation flowshop scheduling problem with earliness and tardiness penalties and common due date of jobs. Based on the constraints imposed by the due dates, we show that the problem can be decomposed into three Sub-problems: one, where the due date is unrestricted, the second, where the due date is restricted, and the third where the due date is in between the restricted and unrestricted due dates. We derive the constraints that categorize the flowshop problems as restricted and unrestricted types.

The solution procedure for all three Sub-Problems presents the first results in the literature that addresses multi-machine problem with E/T penalties. We derive analytical results and obtain optimal solution for Sub-Problem 1 that has unrestricted due date. We propose new heuristics for Sub-Problems 2 and 3 with intermediate and restricted due date respectively. In Sub-Problem 2, for small instances (n = 5, 10; m = 5; I = 50), the average percentage deviation of the heuristic solution from the optimal solution is found to be 0.846 percent and 1.247 percent for 5-jobs and 10-jobs respectively. For large instances, the heuristic solution is compared with the optimal solution obtained at $d = d_1$. The heuristic solution for large problems has very less deviation form optimal solution, with the maximum being 1.744 percent in the case of n = 100, m = 5, I = 50. We discussed that Sub-Problem 3 reduces to that of minimizing tardiness only, and the problem is same as minimizing flowtime or completion time. We compare the performance of the heuristics for solving Sub-Problem 3 with some of the existing results on flowshop tardiness, flowtime, and completion time problems. The proposed heuristics are found to perform better than the existing heuristics.

We have applied these results to schedule finished goods in a large multi-stage production planning and scheduling problem (Chandra, Mehta and Tirupati, 2004). This paper also describes the application of the overall production planning and scheduling problem to a pharmaceuticals company in India with considerable cost savings.

References

Ahmadi, R. H. and U. Bagchi. 1990. Improved Lower Bounds for Minimizing the Sum of Completion Times of *n* Jobs over *m* Machines in a Flowshop. *European Journal of Operational Research*. **44** 331-336.

Bagchi, U., R. Sullivan and Y. Chang. 1986. Minimizing Mean Absolute Deviation of Completion Times about a Common Due Date. *Naval Research Logistics Quarterly*. **33** 227-240.

Baker, K. R. and G. D. Scudder. 1990. Sequencing with Earliness and Tardiness Penalties: A Review. *Operations Research.* **38**(1)**,** 22-36.

Chandra, P., P. Mehta, and D. Tirupati. 2004. Production Planning and Scheduling in a Multi-Stage Batch Processing Environment. Mimeo, Indian Institute of Management, Ahmedabad, India.

Cheng, T.C.E. and H. G. Kahlbacher. 1991. A Proof of the Longest-First Job Policy in One Machine Scheduling. *Naval Research Logistics Quarterly.* **38** 715-720.

Chu, C. 1992. A Branch and Bound Algorithm to Minimize Total Tardiness with Different Release Dates. *Naval Research Logistics Quarterly.* **39** 265-283.

Chu, C. and M. C. Portmann. 1992. Some New Efficient Methods to Solve $n/1/r_i/\sum T_i$ Scheduling Problem. *European Journal of Operation Research*. **58** 404-413.

Gowrishankar, K., C. Rajendran and G. Srinivasan. 2001. Flowshop Scheduling Algorithms for Minimizing the Completion Time Variance and the Sum of Squares of Completion Time deviations from a Common Due Date. *European Journal of Operational Research*. **132** 643-665.

Hall, N., W. Kubiak and S. Sethi. 1991. Earliness-Tardiness Scheduling Problems II: Deviation of Completion Times about a Restrictive Common Due Date. *Operations Research*. **39** 847-856.

Kanet, J. 1981. Minimizing the Average Completion of Job Completion Times About a Common Due Date. *Naval Research Logistics Quarterly.* ***28*** 643-651.

Kanet, J. and V. Sridharan. 2000. Scheduling with Inserted Idle Time: Problem Taxonomy and Literature Review. *Operation Research*. **48** 99-110.

Kim, Y. D. 1995. Minimizing Total Tardiness in Permutation Flowshops. *European Journal of Operational Research*. **85** 541-555.

Koulamas, C. 1994. The Total Tardiness Problem: Review and Extensions. *Operation Research.* **42**(6) 1025-1041.

Nawaz, M and E. I. Ham. 1983. A Heuristic Algorithm for the m-Machine, n-Job Flow-Shop Sequencing Problem. *Omega.* **11** 91-95.

Raghavachari, M. 1986. A V-Shape Property of Optimal Schedule of Jobs about a Common Due Date. *European Journal of Operational Research*. **23** 401-402.

Rajendran, C. 1993. Heuristic Algorithm for Scheduling in a Flowshop to Minimize Total Flowtime. *International Journal of Production Economies*. **29** 65-73.

Rajendran, C. and H. Ziegler. 1997. An Efficient Heuristic for Scheduling in a Flowshop to Minimize Total Weighted Flowtime of Jobs. *European Journal of Operational Research.* **103** 129-138.

Sundararaghavan, P. and M. Ahmed. 1984. Minimizing the sum of absolute lateness in single machine and multi-machine scheduling. *Naval Research Logistics Quarterly.* **31** 325-333.

Taillard, E. 1993. Benchmarks for Basic Scheduling Problems. *European Journal of Operational Research*. **64** 278-285.

Woo, H. S. and D. S. Yim. 1998. A Heuristic Algorithm for Mean Flowtime Objective in Flowshop Scheduling. *Computers and Operations Research*. **25** 175-182.

Appendix 1: Procedure for Generating Alternate Optimal Sequences at $d = d_0$ (GAOS)

The alternate optimal sequences at $d = d_0$ are generated as follows. If the optimal sequence obtained above is index from *1 to n*,

Step 1: $j = 1$

Step 2: $x = j + 1$

Step 3.1: Is $p_{xm} = p_{jm}$

Yes→ Create new sequence by interchanging j and x

$x = x + 1$

is $x = n + 1$

Yes→ $j = j + 1$ and goto step 3.2

No→ repeat step 3.1

No→ $x = x + 1$ and repeat step 3.1

Step 3.2 if $j = n$

STOP else goto step 2

Appendix 2: Procedure for removing idle time at last machine (RIT)

Let the sequence s be 1,2,….n

Step 1: $i = n$

Step 2: $t = S_{im} - C_{i-1m}$

Step 3: If $t > 0$

Yes→ for $x = 1$ to $i-1$

$S_{xm} = S_{xm} + t$

$C_{xm} = S_{xm} + p_{xm}$

If $i = 1$, STOP else

$i = i - 1$ and goto Step 2

No→ If $i = 1$, STOP else

$i = i - 1$ and goto Step 2

In step 1, the last job in the sequence is selected. Step 2 checks if there is an idle time between the jobs. Step 3 removes the idle time between the jobs while maintaining the feasibility of the schedule. This procedure would result in following schedule at machine m.

$C_{nm} = M_{F(s)}$

$S_{nm} = C_{nm} - p_{nm}$

For $i = n\text{-}1$ to 1

$C_{im} = S_{i+1m}$

$S_{im} = C_{im} - p_{im}$

Appendix 3: Heuristic Algorithm for Solving Sub-Problem 2(H1)

Notation

k	=	bottleneck machine
r_{ik}	=	earliest time at which job i is available for processing at machine k
d_{ik}	=	due date of job i at bottleneck machine k
σ	=	a permutation flow shop sequence of n jobs
π	=	set of partial sequence of jobs
$s(\sigma, i)$	=	schedule of sequence σ consisting of S_{ij} and C_{ij} for $\forall i \in \sigma, j = 1,2,\ldots,m$
$Z\{s(\sigma, i)\}$	=	cost of permutation flowshop schedule

$$Z\{s(\sigma, i)\} = \sum_{i=1}^{n} | C_{im} - d |$$

The problem is to determine σ and $s(\sigma, i)$ so as to minimize $Z\{s(\sigma, i)\}$.

<u>Heuristic (H1) for Solving Sub-Problem 2</u>

Step 1 — Determining bottleneck machine k

$$k = \arg\max_{j} \sum_{i=1}^{n} p_{ij}$$

Step 2 — Determining permutation flowshop sequence (σ) and schedule $s(\sigma, i)$ for σ

Step 2.1 — Determining release date of job i at bottleneck machine k

$$r_{ik} = \sum_{x=1}^{k-1} p_{ix} \quad \forall\ i = 1,2,...n$$

Determining due date of job i at bottleneck machine k

$$d_{ik} = d - \sum_{x=k+1}^{m} p_{ix} \quad \forall\ i = 1,2,...n$$

Step 2.2 Determining priority u_i of jobs

$u_i = r_{ik}$ if $r_{ik} + p_{ik} \geq d_{ik}$

$u_i = d_{ik} - p_{ik}$ if $r_{ik} + p_{ik} < d_{ik}$

Step 2.3 Appending a job to π (partial sequence)

Select job with minimum u_i and add to π

Step 2.4 Schedule $s(\pi, i)$ as follows:

for i to $|\pi|$, $i \in \pi$,

for $j = 1$ to m

$S_{11} = 0$

$S_{ij} = \max\ \{C_{ij-1}, C_{i-1j}\}$

$C_{ij} = S_{ij} + p_{ij}$

Step 2.5 Updating r_{ik} $\forall i \notin \pi$

Add i to π and call it π_i

Determine $s(\pi i, i)$ according to step 2.3 $\forall i \in \pi i, j = 1$ to m

$r_{ik} = C_{\pi i\ k-1}$(completion time of i at $(k-1)$ after being appended to π)

This is based on the logic that we schedule the partial sequence π_i according to step 2.4 and determine the time when job i is available for processing at bottleneck machine.

Step 2.6 Updating d_{ik} $\forall i \notin \pi$

$$d_{ik} = \max\ \{d_{ik}, C_{\pi k+1}, \max_{k+2 \leq x \leq m} \{C_{\pi x} - \sum_{y=k+1}^{x-1} p_{iy}\}\}$$

This is based on the logic that a job is not required till the time the partial sequence π is already scheduled on post- bottleneck stages.

Step 2.7 Repeat steps 2.1 to 2.6 for $i \notin \pi$ till $|\Pi| = n$, i.e. a complete sequence σ is obtained.

Step 3 Adjusting the schedule at $j = m$ (last machine)

Shifting all early jobs towards right (increasing C_{im}) before 'd'

Define e: set of early jobs, $e = \{i \mid C_{im} < d\}$

o: set of ontime job: $o = \{i \mid C_{im} = d\}$

t: set of tardy jobs: $t = \{i \mid C_{im} > d\}$

$l = \{i \mid S_{im} < d \text{ and } C_{im} > d\}$

for $i = 1$ to n,

if($C_{im} < S_{i+1m}$ and $C_{im} < d$),

get $z = \min\{S_{i+1m} - C_{im}, d - C_{im}\}$

for $x = 1$ to i

$S_{xm} = S_{xm} + z$

$C_{xm} = C_{xm} + z$

With this all jobs that complete before due date d are shifted towards d so that earliness costs are reduced. This procedure maintains the feasibility of schedule.

Step 4 Improving E/T costs further

if $|e| \geq |o| + |t|$

check if $|o| = 1$

Yes $\rightarrow$ for $i = 1$ to n,

$S_{im} = S_{im} + p_{xm}, x \in o$

$C_{im} = C_{im} + p_{xm}, x \in o$

No $\rightarrow z = d - S_{xm}, x \in l$

for $i = 1$ to n

$S_{im} = S_{im} + z$

$C_{im} = C_{im} + z$

Step 4.1 Bring back (reduce C_{im}) tardy jobs (if they can be) that got shifted towards right after step 4

for $i = 1$ to $|t|$, $i \in t,$

if $C_{im} < S_{i+1m}$ and $S_{i+1m} > C_{im-1}$

Yes → $S_{i+1m} = S_{i+1m} - \min\{S_{i+1m} - C_{im,} S_{i+1m} - C_{im-1}\}$

$C_{i+1m} = S_{i+1m} + p_{i+1m}$

No → $S_{i+1m} = S_{i+1m}$

$C_{i+1m} = C_{i+1m}$

Step 5 Determine $Z\{s(\sigma, I)\} = \sum_{i=1}^{n} | C_{im} - d|$

Step 6 Improving the objective value by performing neighbor hood search scheme (tabu search) to get a better sequence and schedule. The tabu search procedure is described below.

Tabu Search Procedure (TS)

Z_c	=	objective function of the current best solution
σ_c	=	current best sequence
Z_e	=	objective function of the best ever solution
σ_e	=	best ever sequence
p	=	number of pairs, $p = n(n-1)/2$
t	=	number of tabu iterations
Z_{xj}	=	objective function of the candidate sequence formed by interchanging j^{th} pair, $j = 1,2,\ldots p$
σ_{xj}	=	sequence of candidate sequence x formed by interchanging j^{th} pair, $j = 1,2,\ldots p$.
a_j	=	$Z_c - Z_{xj}$,
ts_j	=	tabu structure of the j^{th} pair, $0 \leq ts_j \leq$ tabu tenure

Step 6.1 for $i = 1$ to t

Step 6.1.1 for $j = 1$ to p

Generate p candidate sequences σ_{xj} by interchanging j^{th} pair from the current best sequence σ_c, $x = 1,2,\ldots p$

Schedule the sequence x from step 2.4, step 3 and step 4.

Determine Z_{xj} from step 5

Determine $a_j = Z_c - Z_{xj}$

Sort d_j*'s* in non-increasing order and re-index d_j from 1 to n

Step 6.2 $j = 1$

Step 6.3

Case 1: Candidate solution is worse than current solution and the pair is tabu as well

$a_j \le 0$ and $ts_j > 0$

$j = j+1$ and repeat step 6.3

Case 2: Candidate solution is better than current solution and the pair is not tabu

if $a_j > 0$ and $ts_j = 0$

step 6.3.1 $Z_c = Z_{xj}$

$\sigma_c = \sigma_{xj}$

$ts_j =$ tabu tenure

for $j = 1$ to p

if $ts_j > 0$

$ts_j = ts_j - 1$

if $Z_c < Z_e$

$Z_e = Z_c$

$\sigma_e = \sigma_c$

Case 3: Candidate solution is worse than the current solution and the pair is tabu

if $a_j \le 0$ and $ts_j = 0$ goto step 6.3.1

Case 4: Candidate solution is better than the current solution, better than best ever solution but the pair is tabu (Aspiration)

if $a_j > 0$ and $ts_j > 0$ and $Z_e > d_j$

goto step 6.3.1

Step 6.4 If $i = t$, STOP, else $i = i + 1$ and goto Step 6.1.1.

Appendix 4: Heuristic Algorithm (H2) for Sub-Problem 3

Steps 1 to steps 2.1 are same as in Heuristic H1 for solving Sub-Problem 2.

Step 2.2 Determining priority u_i of jobs

$$u_i = max(r_{ik}, t) + max\{max(r_{ik}, t) + p_{ik}, d_{ik}\}$$

where t = current time = $C_{\sigma k}$

Step 2.3 to step 2.7 are same as in Heuristic H1

Steps 3 and steps 4 are not required as no job is early.

Steps 5 and steps 6 are same as in Heuristic H1