The CON Job Scheduling Problem on a Single and Parallel Machines

Huajun Zhou

A Thesis Submitted to the
University of North Carolina at Wilmington in Partial Fulfillment
Of the Requirements for the Degree of
Master of Science

Department of Mathematics and Statistics

University of North Carolina at Wilmington

2003

Approved by

Advisory Committee

_____ _____

_____

Chair

Accepted by

_____

Dean, Graduate School

TABLE OF CONTENTS

ABSTRACT

We study job scheduling problems on both a single machine and on independent identical parallel machines. We impose a due date cost and each of the jobs have earliness and tardiness costs. The single machine problem considers that all the jobs are processed by only one machine. The goal is to find the optimal due date that minimizes the cost. The parallel machines problem considers m identical machines which process the jobs simultaneously. We consider several variations of the problem. These include the constant flow allowance problem, job independent and job dependent cost problems, and the slack due date problem. Our main contribution is a new algorithm for job dependent costs on parallel machines.

ACKNOWLEDGMENTS

I would like to express my sincere appreciation to Dr. John K. Karlof for guidance on my thesis work. I also want to thank Dr.Wei Feng and Dr. Yaw O. Chang for serving on my thesis advisory committee and offering constructive ideas and all faculty members in the Mathematics and statistics Department who gave me valuable advice and help. I also want to thank my husband Lin Cheng for his patience and help.

This thesis is dedicated to my parents.

1 INTRODUCTION

The subject of scheduling jobs on machines has attracted much research attention over the years. In this paper, we study job scheduling problems on both single machine and on independent identical parallel machines. There is a due cost and each of the jobs have earliness and tardiness costs. The single machine problem considers that all the jobs are processed by only one machine. The goal is to find the optimal due date that minimize the cost. The parallel machines problem considers m identical machines which process the jobs simultaneously. We consider several variations of the problem. These include the constant flow allowance problem, job independent and job dependent costs, and the slack due date problem. Our main contribution is a new algorithm for job dependent costs on parallel machines.

Definition 1 *Constant Flow Allowance problem(CON problem) : the due date of each job is the same.*

Definition 2 *Job-dependent problem : each job has its own due date cost, earliness cost and tardiness cost which is assigned before the job sequence is determined.*

Definition 3 *Job-independent problem : all the jobs have identical due date costs, earliness costs and tardiness costs which are assigned before the job sequence is determined.*

The CON problem on a single machine with job dependent costs was studied by Prabuddha De, Jay B. Ghosh,Charles E. Wells in [1]. The problem was modelled as a $0-1$ quadratic programming problem. The CON problem on a single machine with job independent costs was studied by S.S. Panwalker and M.L. Smith [2]. A labelling algorithm was presented that produced the optimal job sequence.

Definition 4 *Slack due date problem (SLK problem) : the n jobs are each assigned*

a different due date. The due date for job i, d_i, equals the sum of the job's processing time p_i plus a common flow allowance q, that is

$$d_i = p_i + q.$$

The SLK problem on a single machine with job independent costs was presented by by George I. Adamopoulos and Costas P. Pappis [3]. An algorithm with 5 steps to find the optimal solutions was introduced.

In section 2 we present the single machine scheduling problem. We introduce both the job dependent CON problem and the job independent CON problem. The single machine problem is the basis of the more complicated parallel machine problem. In the single machine scheduling problem, we introduce the V-shape property along with some other properties of the optimal sequence. In section 3 we present the parallel machine scheduling problem. We first introduce the job-independent problem and a heuristic algorithm and program it in MATLAB. Then we derive an algorithm for the job-dependent problem where the due date cost, earliness cost and the tardiness cost are totally different for each job. Finally we give a numerical example to show the above method step by step.

2 Single Machine Scheduling

We first introduce the problem of finding an optimal schedule on a single machine. We consider n independent jobs that are immediately available for continuous processing on a single machine. Each job has its own processing time. Costs are imposed for:

- a specific due date,
- tardiness,
- earliness.

The objective is to find a job sequence that minimizes the total cost.There are three different classes of problems.

2.1 Constant Flow Allowance -CON

The due date of the *constant flow allowance* (CON) problem is the same for all jobs. This due date can be set in two ways. On the one hand, the customer decides the due dates for all the jobs or on the other hand, the production company sets the due date in consultation with the customers. We use the following terminology in the CON problem:

J_i**:** the job in position i in S,

S**:** an arbitrary sequence of jobs,

d_i**:** the due date of job J_i in S,

k**:** constant.

Thus

$$d_i = k \text{ for all } J_i \in S.$$

In the constant flow allowance problem, we consider both job-independent and job-dependent due date costs, tardiness costs and earliness costs. We allow the tardiness cost and the earliness cost to be different. Usually we treat d_i as d since each job has the same due date in the following cases. Now we introduce two types of CON problems.

2.1.1 Job-dependent CON problem

In this problem we consider that the due date, earliness/tardiness costs are job-dependent and denoted by D_i, e_i, t_i. Thus

$$E_i = \max\{0, d - c_i\},\ T_i = \max\{0, c_i - d\}$$

where

d**:** common due date,

c_i**:** completion time of job i,

E_i**:** job-dependent earliness for job i,

T_i**:** job-dependent tardiness for job i.

Property 1 below was proved by M. A. Quaddus [4] and concerns finding the optimal due date d^*.

Property 1 *For any specified sequence S, there exists an optimal due-date* d^* *where* $d^* = c_r$ *and* r *is the smallest integer such that*

$$\sum_{i=1}^{r}(e_i + t_i) \geq \sum_{i=1}^{n}(t_i - D_i)$$

where

D_i**:** *due date cost for job i,*

e_i**:** *job-dependent earliness cost for job i,*

t_i**:** *job-dependent tardiness cost for job i.*

We outline M. A. Quaddus' proof.

Proof: The linear programming(LP) model of the job-dependent CON problem is as follows:

$$\min f(d) = \sum_{i=1}^{n}(D_i d + e_i E_i + t_i T_i)$$

Subject to :

$$\begin{aligned} d + T_i - E_i &= c_i \\ d, T_i, E_i &\geq 0 \end{aligned}$$

Let Y=$(y_1, y_2, ...y_n)$ be the vector of dual variables. The dual problem then becomes:

$$\max\ g(Y) = \sum_{i=1}^{n} c_i y_i$$

Subject to :

$$\begin{aligned} \sum_{i=1}^{n} y_i &\leq \sum_{i=1}^{n} D_i \\ y_i &\leq t_i \\ -y_i &\leq e_i \end{aligned}$$

$$y_i \text{ unrestricted}$$

Let r and d^* be as defined in the statement of property 1. Define the following dual

variables:

$$\begin{aligned} {y_i}^* &= -e_i,\ 1 \le i \le r-1 \\ {y_i}^* &= t_i,\ r+1 \le i \le n \\ \text{and } {y_r}^* &= \sum_{i=1}^{r-1} e_i - \sum_{i=r+1}^{n} t_i + \sum_{i=1}^{n} D_i \end{aligned}$$

By making use of the inequality of property 1, it can be shown that $y^* = (y_1^*, y_2^*, ..., y_n^*)$, as defined above, is feasible. Now we show that $f(d^*) = g(Y^*)$. The objective function $g(Y^*)$ of the dual problem is

$$\begin{aligned} g(Y^*) &= -\sum_{i=1}^{r-1} c_i e_i + \sum_{i=r+1}^{n} c_i t_i + \sum_{i=1}^{r-1} c_i e_i \\ &\quad - \sum_{i=r+1}^{n} c_i t_i + \sum_{i=1}^{n} c_i D_i \\ &= \sum_{i=1}^{n} c_i D_i \end{aligned}$$

And the objective function of the primal problem can be gotten by ${y_i}^* b$ where b means the be the column vector of primal variables. So

$$\begin{aligned} f(d^*) &= -\sum_{i=1}^{r-1} c_i e_i + \sum_{i=r+1}^{n} c_i t_i + \sum_{i=1}^{r-1} c_i e_i \\ &\quad - \sum_{i=r+1}^{n} c_i t_i + \sum_{i=1}^{n} c_i D_i \\ &= \sum_{i=1}^{n} c_i D_i \end{aligned}$$

Thus $f(d^*) = g(Y^*)$ which completes the proof.■

Property 2 can be found in [1]. It concerns the optimal job sequence.

Property 2 *In an optimal sequence S, for jobs $j, k \in E$, j precedes k if $p_j/e_j \ge p_k/e_k$; similarly for jobs $j, k \in T$, j follows k if $p_j/t_j \ge p_k/t_k$*

where

E, T**:** *sets of early and tardy jobs,*

p_j, p_k**:** *processing time of job j,k,*

e_j, e_k**:** *job-dependent earliness cost of job j,k,*

t_j, t_k**:** *job-dependent tardiness cost of job j,k.*

We now include our proof of Property 2:

Proof: Assume job j precedes job $k \in E$, common due date d, processing time p_j of job j and p_k for job k, job-dependent earliness cost e_j of job j and e_k of job k, and the due date cost z. We first assume the completion time c_j of job j is a constant β (the total time of the jobs proceeding job j) plus processing time p_j. And the completion time $c_k = \beta + p_j + p_k$ since job j precedes job k. Thus the total cost is

$$\begin{aligned} & z + e_j(d - c_j) + e_k(d - c_k) \\ = \; & z + e_j(d - (p_j + \beta)) \\ & + e_k(d - (p_j + p_k + \beta)) \\ = \; & z + e_j d - e_j p_j - e_j \beta \\ & + e_k d - e_k p_j - e_k p_k - e_k \beta \end{aligned} \tag{1}$$

Then if job k precedes job j,the total cost is

$$\begin{aligned} & z + e_j(d - c_j) + e_k(d - c_k) \\ = \; & z + e_j(d - (p_k + p_j + \beta) \\ & + e_k(d - (p_k + \beta)) \\ = \; & z + e_j d - e_j p_k - e_j p_j \\ & - e_j \beta + e_k d - e_k p_k - e_k \beta \end{aligned} \tag{2}$$

If the sequence is optimal and job j proceeds job k, (1) must be less than or equal to (2).Thus

$$\begin{aligned} & z + e_j d - e_j p_j + e_k d - e_k p_j - e_k p_k \\ \leq \; & z + e_j d - e_j p_k - e_j p_j + e_k d - e_k p_k. \end{aligned}$$

Then we get

$$-e_k p_j \leq -e_j p_k \ .$$

And

$$\frac{p_j}{e_j} \geq \frac{p_k}{e_k}$$

Similarly, we can prove that for jobs $j, k \in T$, j follows k if $p_j/t_j \geq p_k/t_k$. ■

Methods to solve these kinds of problems were introduced by Prabuddha De, Jay B. Ghosh and Charles E. Wells[1]. They cast the problem as an 0-1 quadratic programming problem and solved it using Lindo. They used a randomized adaptive search procedure(GRASP) to do the computational study. At the end of the paper, they provide several different results from different kinds of methods and verify the GRASP's effectiveness. The more the iterations, the better the results. The quality

of the heuristic remains to be tested for larger problem sizes. Since obtaining an optimal solution will be near impossible in such instances, the focus needs to be directed towards the identification of good lower bounds on the value of the optimal solution. It will also be interesting to explore certain obvious modifications to the heuristic.

2.1.2 Job-independent CON problem

In this problem we consider the earliness costs and tardiness costs are job-independent. This means we specify the due date assignment cost per unit time P_1, the earliness cost per unit time P_2, and the tardiness cost per unit time P_3 independent of which job is processing on the machine [2]. We define the SPT sequence as a non-decreasing jobs' sequence by processing time t_i as $t_1 \leq t_2 \leq ... \leq t_n$. The following property 3 and lemma 1 are found in [2] without proof. We outline our own proofs.

Property 3 *If* $P_1 \geq P_3, d^* = 0$ *and the sequence is optimal.*

Proof:

$$\min f(d) = \sum_{i=1}^{n}(P_1 d + P_2 E_i + P_3 T_i)$$

Subject to :

$$d + T_i - E_i = c_i$$
$$d, T_i, E_i \geq 0 \quad .$$

Let Y=$(y_1, y_2, ...y_n)$ be the vector of dual variables. The dual problem then becomes:

$$\max \ g(Y) = \sum_{i=1}^{n} c_i y_i$$

Subject to :

$$\sum_{i=1}^{n} y_i \leq \sum_{i=1}^{n} P_1$$

$$y_i \leq P_3$$

$$-y_i \leq P_2$$

$$y_i \text{ unrestricted.}$$

Assume that ${y_i}^* = P_3$. It can be seen that ${y_i}^*$ is feasible as it satisfies all the constraints of the dual problem. We show that the objective functions of the primal and dual problem have the same value. The objective function value of the dual problem is

$$g(Y^*) = \sum_{i=1}^{n} c_i P_3.$$

The primal solution associated with $d^* = 0, E_i^* = 0$ and $T_i^* = c_i$ is

$$f(d^*) = \sum_{i=1}^{n} c_i P_3.$$

Thus $f(d^*) = g(Y^*)$ which completes the proof. ■

So for the remainder of this section, we assume $P_1 \leq P_3$.

Lemma 1 *For any specified sequence,there exists an optimal due date equal to c_k,where k is the smallest integral value greater than or equal to $\frac{n(P_3-P_1)}{(P_2+P_3)}$.*

Proof: The objective function can be written as $f(d,S) = nP_1d + P_2\sum_{i=1}^{r-1}(d - c_i) + P_3\sum_{i=r+1}^{n}(c_i - d)$. After simplifying, we obtain $f(d,S) = (nP_1 + (r-1)P_2 - (n-r)P_3)d - P_2\sum_{i=1}^{r-1} c_i + P_3\sum_{i=r+1}^{n} c_i$. For this linear function, the slope is $nP_1 + (r-1)P_2 - (n-r)P_3$, y-intercept is $-P_2\sum_{i=1}^{r-1} c_i + P_3\sum_{i=r+1}^{n} c_i$ and x-intercept is $\frac{(-P_2\sum_{i=1}^{r-1} c_i + P_3\sum_{i=r+1}^{n} c_i)}{(nP_1+(r-1)P_2-(n-r)P_3)}$. The minimum value of a linear function that has positive slope is the y-intercept. In this case, $d^* = 0$. From the proof of property 3 we know that if $d^* = 0$ this sequence is optimal. But this result is trivial in our research. For the linear objective function which has negative slope, the domain of the due

date d is $(0, \frac{(-P_2 \sum_{i=1}^{r-1} c_i + P_3 \sum_{i=r+1}^{n} c_i)}{(nP_1+(r-1)P_2-(n-r)P_3)})$. Thus so that the x-intercept is great than 0 we obtain $-P_2 \sum_{i=1}^{r-1} c_i + P_3 \sum_{i=r+1}^{n} c_i \leq 0$. But it is impossible in this case for the job-independent CON problem to have negative y-intercept. Thus we know that the objective function must have slope 0 which implies

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

So r is given by

$$r = \frac{n(P_3-P1)}{P_2+P_3} + \frac{P_2}{P_2+P_3}.$$

Since we already know that all jobs before position r(including this position) are early and all jobs after the position are tardy, the objective function can be written as

$$f(d,S) \quad = \quad nP_1 d + P_2 \sum_{i=1}^{r} (d - c_i) + P_3 \sum_{i=r+1}^{n} (c_i - d).$$

It is obvious that r must be the smallest integer not less than $r = \frac{n(P_3-P1)}{P_2+P_3}$. Thus

$$r = \lceil \frac{n(P_3-P1)}{P_2+P_3} \rceil \quad \blacksquare$$

Thus, also in this case, we know that an optimal value of d^* coincides with the completion time of a job. One method to solve job-independent CON problems is introduced by S.S. Panwalker and M.L. Smith [2]. In that paper, they present a polynomial bound scheduling algorithm for the solution. The algorithm presented consists of two phases. Step 1 to step 3 finds the number of nontardy jobs(r). Step 4 to step 7 calculates positional labels γ, the optimal sequence, and the optimal due date d^*.

Step 1:

Set $r' = \frac{n(P_3-P1)}{P_2+P_3}$

Step 2:

Check if $r' > 0$. If Yes: go to next step,

If No: set $d^* = 0$, and the sequence is optimal.

Step 3:

Let $r = \lceil r' \rceil = \lceil \frac{n(P_3 - P_1)}{(P_2 + P_3)} \rceil$.

Step 4:

Set label position $j (1 \leq j \leq n)$ as

$$\gamma = \begin{cases} nP_1 + (j-1)P_2, & 1 \leq j \leq r \\ (n+1-j)P_3, & r+1 \leq j \leq n. \end{cases}$$

Step 5:

Rank the positional labels in descending order such that the largest γ is ranked 1 and the smallest γ is ranked n. Break ties arbitrarily.

Step 6:

Obtain the optimal sequence such that job i is scheduled in position j corresponding to γ ranked in position i.

Step 7:

Set $d^* = t_1 + t_2 + ... + t_r$.

Thus each position j has a fixed position label. Arrange the position j by increasing order with the position labels and then rank the labels. Thus the objective function $f(d)$ is minimized by matching the smallest value of lable γ with the largest value of j, the next larger value of γ with the next smaller value of t, and so on.

2.2 Slack Due Date -SLK

In the slack due date problem, the n jobs are each assigned a different due date. The due date for job i, d_i, equals the sum of the job's processing time plus a common flow allowance, that is

$$d_i = p_i + q.$$

The following lemma shows that, in any sequence, there is a job occupying position r such that any job after this is tardy and any job before this is early. This is proved by George I. Adamopoulos and Costas P. Pappis [3].

Lemma 2 *If $c_i \geq d_i$, then $c_{i+1} \geq d_{i+1}$. Similarly if $c_i \leq d_i$, then $c_{i+1} \leq d_{i+1}$. where*

c_i: completion time of job at position i,

d_i: the due date of job at position i.

Proof: Let S be a given sequence. Since we know that

$$c_{i-1} + p_i = c_i \quad \text{and} \quad p_i + q = d_i$$

then from the lemma 2 we have

$$c_{i-1} + p_i \;\geq\; p_i + q$$

then

$$\begin{aligned} c_{i-1} &\geq q \\ c_{i-1} + p_i &\geq q \\ c_i &\geq q \\ c_i + p_{i+1} &\geq p_{i+1} + q \\ c_{i+1} &\geq d_{i+1}. \blacksquare \end{aligned}$$

Theorem 1 *The optimal position r is given by the smallest integer not less than* $\frac{n(P_3-P_1)}{(P_2+P_3)}$.

Proof:

$$\begin{aligned} f(q,S) &= \sum_{i=1}^{n}[P_1 q + P_2 E_i + P_3 T_i] \\ &= nP_1 q + \sum_{i=1}^{n}[P_2 E_i + P_3 T_i] \\ &= nP_1 q + \sum_{i=1}^{n}[P_2(d_i - C_i) + P_3(C_i - d_i)] \\ &= nP_1 q + \sum_{i=1}^{r-1} P_2(q + t_i - C_i) + \sum_{i=r+1}^{n} P_3(C_i - q_i - t_i) \\ &= (nP_1 + (r-1)P_2 - (n-r)P_3)q + \sum_{i=1}^{r-1} P_2(t_i - C_i) + \sum_{i=r+1}^{n}(C_i - t_i). \end{aligned}$$

For the above linear objective function, we obtain the minimum value when the slope is zero. The reasoning is similar to the proof of lemma 1. Thus

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

So r is given by

$$r = \tfrac{n(P_3-P1)}{P_2+P_3} + \tfrac{P_2}{P_2+P_3}.$$

Therefore

$$r = \lceil \tfrac{n(P_3-P1)}{P_2+P_3} \rceil. \blacksquare$$

A method to solve the slack due date problem was introduced by George I. Adamopoulos and Costas P. Pappis [3]. In that paper they provide an algorithm with 5 steps to find the optimal solutions.

Step 1:

Determine the optimal position r.

Step 2:

Calculate the position labels and waiting time.

Step 3:

Allocate the jobs in positions.

Step 4:

Arrange the position labels sequence as an non-decreasing order and processing times sequence as an non-increasing order.

Step 5:

Evaluate the objective function and terminate.

3 Parallel Machine Scheduling

In section 2, we introduced the Single Machine Scheduling problem of finding an optimal schedule on a single machine. In this section, we are concerned with the due-date assignment and early/tardy scheduling on identical parallel machines. We consider n independent jobs that are immediately available for continuous processing on m identical parallel machines ($m \leq n$) and a job cannot be preempted once its processing has begun. Our goal is to assign a common due date d and a schedule of jobs on the parallel machines that minimizes the total cost.

3.1 Heuristic Solution of Job-independent Problem

We use the following terminology in the parallel machine problem with job independent costs:

d**:** the common due date for all the jobs,

$C_{j[i]}$**:** completion time of the job in position i on machine j,

$E_{j[i]}/T_{j[i]}$**:** earliness and tardiness of the job in position i on machine j,

P_1**:** the due date assignment cost per unit time,

P_2/P_3**:** the earliness/tardiness cost per unit time.

Thus

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

and the objective is to minimize a penalty function given by

$$\begin{aligned} f(d,S) &= \sum_{j=1}^{m}\sum_{i=1}^{n}(P_1 d + P_2 E_{j[i]} + P_3 T_{j[i]}) \\ &= \sum_{j=1}^{m}\sum_{i=1}^{n}[P_1 d + P_2(d - C_{j[i]}) + P_3(C_{j[i]} - d)] \\ &= \sum_{j=1}^{m}[nP_1 d + P_2\sum_{i=1}^{r-1}(d - C_{j[i]}) + P_3\sum_{i=r+1}^{n}(C_{j[i]} - d)] \\ &= \sum_{j=1}^{m}((nP_1 + (r-1)P_2 - (n-r)P_3)d - P_2\sum_{i=1}^{r-1}C_{j[i]} + P_3\sum_{i=r+1}^{n}C_{j[i]}) \quad (3) \end{aligned}$$

The procedure to find the optimal due-date value for the parallel-machine problem can be derived in a similar way as what we did in lemma 1 and theorem 1. The above linear function must have slope 0. Thus, we set

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

to obtain the optimal value. For a given job sequence S, let the jobs in S be indexed according to non-decreasing completion time, ie. $C_1 \leq C_2 \leq C_3 ... \leq C_n$. If the

optimal due-date is set as $d^* = C_r$, then the above equation requires r to satisfy the following condition

$$r = \frac{n(P_3 - P1)}{P_2 + P_3} + \frac{P_2}{P_2 + P_3}$$

which is identical to the results of lemma 1 of the single machine scheduling problem. It is clear, in this case, that $d^* = C_r$ is dependent on the job sequence S. But in the single-machine case, $d^* = C_r$ is a function of the job position r that is independent of the sequence S. This is the similarity and difference between the single-machine problem and parallel-machine problem.

A method to solve the parallel machine problem was introduced by T. C. E. Cheng [5]. In that paper, he wasn't aware of the existence of either a polynomial-bound algorithm or a heuristic procedure to solve the parallel machine problem. He proposed a simple heuristic based on the optimal results derived from the single machine problem.

3.1.1 Heuristic Solution

In this method he specifies the due date assignment cost per unit time P_1, the earliness cost per unit time P_2, and the tardiness cost per unit time P_3 independent of which job is processing on the different machine. Here we outline T. C. E. Cheng's 9 step algorithm.

Step 1:

Construct a list S of jobs 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 $\lceil x \rceil$ is the smallest integer greater than or equal to x. In this step, we decide how many jobs can at most be assigned to each machine.

Step 3:

Set the optimal due date for each machine $r = \lceil \frac{k(P_3 - P_1)}{(P_2 + P_3)} \rceil$.

Step 4:

Calculate the position labels in this step. Set $\lambda_{j[i]} = kP_1 + (i-1)P_2$ for $1 \leq i \leq r$ and $1 \leq j \leq m$; $\lambda_{j[i]} = (k+1-i)P3$ for $r+1 \leq i \leq k$ and $1 \leq j \leq m$.

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 job and the first position label from S and A respectively. Go to step 6 till set S and set A are empty.

Step 8:

A job sequence s is generated. Calculate the completion times of the jobs in s and re-index the jobs in non-decreasing order C_i, such that $C_1 \leq C_2 \leq C_3 ... \leq C_n$. Set $d^* = C_r$, where $r = \lceil \frac{n(P_3 - P_1)}{(P_2 + P_3)} \rceil$.

Step 9:

Evaluate

$$f(d^*, s) = \sum_{j=1}^{m} \sum_{i=1}^{n} \{P_1 d^* + P_2 E_{j[i]} + P_3 T_{j[i]}\}$$

The heuristic first constructs a list S, of jobs in non-decreasing processing times(step 1), and then calculates k, the average number of jobs per machine(step 2). The algorithm then treats each machine individually as though they are m independent

single-machine problems. Applying the optimal results of the single-machine problem, the heuristic finds the optimal position r(step 3), and defines a position label for each job on each individual machine(step 4). The next three steps(step 5,6 and 7) are followed to assign jobs to machine positions in such a way as to match the longest jobs with the position label having the smallest value, the second longest job with the second smallest label, and so on until the set S is used up. This results in a job sequence s, from which the job completion times are identified and d^* determined, the optimal due-date for the parallel-machine problem. Once the due-date value and job sequence are determined, the corresponding penalty function value $f(d^*, s)$ can be evaluated(step 9).

3.1.2 A Numerical Example

To demonstrate the working procedure of the heuristic method, consider the problem of scheduling $n = 9$ jobs on $m = 3$ machines. The job processing time are given as $p_1 = 1, p_2 = 2, p_3 = 3, p_4 = 4, p_5 = 5, p_6 = 6, p_7 = 7, p_8 = 8, p_9 = 9$. The penalty costs are estimated as $P_1 = 1, P_2 = 2, P_3 = 3$. Applying the heuristic method yields the following results:

Step 1:

$S := 1, 2, 3, 4, 5, 6, 7, 8, 9$ and $|S| := 9$.

Step 2:

Set $k = \lceil \frac{9}{3} \rceil = 3$.

Step 3:

Set $r = \lceil \frac{3\times(3-1)}{2+3} \rceil = 2$.

Step 4:

Set

$$\lambda_{j[i]} := \begin{cases} 1+2i, & 1 \le i \le 2, 1 \le j \le 3 \\ (4-i)3, & i = 3, 1 \le j \le 3. \end{cases}$$

Step 5:

$A = \{\lambda_{1[2]} = 5, \lambda_{2[2]} = 5, \lambda_{3[2]} = 5, \lambda_{1[1]} = 3, \lambda_{2[1]} = 3, \lambda_{3[1]} = 3, \lambda_{1[3]} = 3, \lambda_{2[3]} = 3, \lambda_{3[3]} = 3\}$.And$|A| := 9$.

Step 6 and step 7:

$s = \{1[1] = 4, 1[2] = 1, 1[3] = 7, 2[1] = 5, 2[2] = 2, 2[3] = 8, 3[1] = 6, 3[2] = 3, 3[3] = 9\}$.

Step 8:

$C_1 = 4,\ C_2 = 5,\ C_3 = 5,\ C_4 = 6,\ C_5 = 7,\ C_6 = 9,\ C_7 = 12,\ C_8 = 15,\ C_9 = 18.$

$r = \lceil \frac{9\times(3-1)}{2+3} \rceil = 4$

And $d^* = C_4 = 6$.

Step 9:

Evaluate

$f(d^*, s) = \sum_{j=1}^{m} \sum_{i=1}^{n} \{P_1 d^* + P_2 E_{j[i]} + P_3 T_{j[i]}\}$

$= \{9 \times 1 \times 6 + 2 \times 2 + 2 \times 1 + 2 \times 1 + 3 \times 1 + 3 \times 3 + 3 \times 6 + 3 \times 9 + 3 \times 12\}$

$= 155.$

3.1.3 Improvement on Heuristic Solution

From the above heuristic solution, it restricts the optimal solution to only the non-delay schedules. Then Kenneth R. Baker and Gary D. Scudder [6] pointed out that the optimal start time may be nonzero and the sufficient condition for the optimal

Figure 1: Job-independent Heuristic Solution

schedule to start at time zero is

$$p_1 + p_2 + ... + p_i \geq d$$

where

$$i = \frac{n-1}{2}, \text{ if n is odd and}$$
$$i = \frac{n}{2}, \text{ if n is even.}$$

Since $p_1 = 1 \leq d^* = 11$ from the numerical example above, the heuristic solution above is a near-optimal method. We can assume that the start time of the schedule is nonzero. An improvement of the heuristic solution was introduced by Prabuddha De, Jay B. Ghosh and Charles E. Wells [7]. They proposed that the optimal start time be identified as following:

Step 1:

Assume start times at zero and determine the optimal due-date d^* as described by the heuristic solution.

Step 2:

For machine $j, j = 1, 2, 3, ..., m$, compute $r_j = \lceil \frac{n_j P_3}{P_2 + P_3} \rceil$, where n_j is the number of jobs in machine j.

Step 3:

If the completion time of the rth job in machine j, $C_{j[r]}$, is less than d^*, then $\delta_j = d^* - C_{j[r]}$, otherwise $\delta_j = 0$.

Step 4:

Delay the first job of machine j by δ_j units. And correspondingly change the completion time of the following jobs processed through machine j.

Step 5:

Evaluate

$$f(d^*, s) = \sum_{j=1}^{m} \sum_{i=1}^{n} \{P_1 d^* + P_2 E_{j[i]} + P_3 T_{j[i]}\}.$$

To illustrate the point, we use the numerical example from above, which considers the problem of scheduling 9 jobs on 3 machines. The job processing times are given as $p_1 = 1, p_2 = 2, p_3 = 3, p_4 = 4, p_5 = 5, p_6 = 6, p_7 = 7, p_8 = 8, p_9 = 9$. The penalty costs are estimated as $P_1 = 1, P_2 = 2, P_3 = 3$. We calculate the total penalty of 155 by using the heuristic solution. Now we use the improvement of the heuristic solution to get the new optimal penalty cost.

Step 1:

set all start times and to be zero and determine the optimal due-date $d^* = 6$ by the heuristic solution.

Step 2:

For each machine , $r_1 = \lceil \frac{3\times3}{5} \rceil = 2$, $r_2 = \lceil \frac{3\times3}{5} \rceil = 2$, $r_3 = \lceil \frac{3\times3}{5} \rceil = 2$.

Step 3:

$C_{1[2]} = 5 < 6,\ \delta_1 = 6 - 5 = 1,$

$C_{2[2]} = 7,\ \delta_2 = 0,$

$C_{3[2]} = 9,\ \delta_3 = 0.$

Step 4:

Delay the first job of machine 1 by 1 units. And $C_1 = 5,\ C_2 = 5,\ C_3 = 6,\ C_4 = 6,\ C_5 = 7,\ C_6 = 9,\ C_7 = 13,\ C_8 = 15,\ C_9 = 18.$

Step 5:

Evaluate $f(d^*, s)$

$\sum_{j=1}^{m} \sum_{i=1}^{n} \{P_1 d^* + P_2 E_{j[i]} + P_3 T_{j[i]}\}$

$= \{9 \times 1 \times 6 + 2 \times 1 + 2 \times 1 + 3 \times 1 + 3 \times 3 + 3 \times 7 + 3 \times 9 + 3 \times 12\}$

$= 154.$

Figure 2: Job-independent Improvement Heuristic Solution

3.2 Job-Dependent Problem for Parallel Machines

In sections 3.1 and 3.2, we considered due date costs, earliness costs, and tardiness costs that were job-independent. In this section, we consider the above variables as job-dependent on parallel machines.

3.2.1 Heuristic Solution

Property 4 *For any specified sequence S on machine j, there exists an optimal due-date d_j^* for machine j and $d_j^* = c_{r_j}$ where r_j is the smallest integer such that*

$$\sum_{i=1}^{k} D_{j[i]} \leq \sum_{i=r_j+1}^{k} t_{j[i]}$$

where

$D_{j[i]}$**:** *due date cost for job i on machine j,*

k**:** *number of jobs on each machine where* $k = \lceil n/m \rceil$*, n is the number of jobs and m is the number of machines,*

$t_{j[i]}$**:** *job-dependent tardiness cost for job i on machine j.*

Proof: The linear programming(LP) model of the job-dependent parallel machine problem is as follows:

$$\begin{aligned} f(d,S) &= \sum_{j=1}^{m}\sum_{i=1}^{k}(D_{j[i]}d_j + e_{j[i]}E_{j[i]} + t_{j[i]}T_{j[i]}) \\ &= \sum_{j=1}^{m}\sum_{i=1}^{k}(D_{j[i]}d_j + e_{j[i]}(d_j - c_{j[i]}) + t_{j[i]}(c_{j[i]} - d_j)) \\ &= \sum_{j=1}^{m}[d_j\sum_{i=1}^{k}D_{j[i]} + \sum_{i=1}^{r_j-1}e_{j[i]}(d_j - c_{j[i]}) + \sum_{i=r_j+1}^{k}t_{j[i]}(c_{j[i]} - d_j)]. \end{aligned}$$

For the above parallel machine problem if the value of the objective function for each machine is optimal, then the whole objective function value is surely optimal. Thus we now consider the objective function of any one arbitrary machine q as f' where

$$\begin{aligned} f'(d,S) &= d_q\sum_{i=1}^{k}D_{q[i]} + \sum_{i=1}^{r_q-1}e_{q[i]}(d_q - c_{q[i]}) + \sum_{i=r_q+1}^{k}t_{q[i]}(c_{q[i]} - d_q) \qquad (4) \\ &= (\sum_{i=1}^{k}D_{q[i]} + \sum_{i=1}^{r_q-1}e_{q[i]} - \sum_{i=r_q+1}^{k}t_{q[i]})d_q - \sum_{i=1}^{r_q-1}e_{q[i]}c_{q[i]} + \sum_{i=r_q+1}^{k}t_{q[i]}c_{q[i]}. \qquad (5) \end{aligned}$$

We already know that the above linear function has the optimal value at slope 0 . Then

$$\sum_{i=1}^{k} D_{q[i]} + \sum_{i=1}^{r_q-1} e_{q[i]} - \sum_{i=r_q+1}^{k} t_{q[i]} = 0.$$

Furthermore,

$$\sum_{i=1}^{k} D_{q[i]} \leq \sum_{i=r_q+1}^{k} t_{q[i]}.$$

Thus

$$\sum_{i=1}^{k} D_{j[i]} \leq \sum_{i=r_j+1}^{k} t_{j[i]}. \blacksquare$$

From the above property, we can see that the total due date costs for each machine is always less or equal to the total tardiness costs after the optimal due date is chosen. We derive a scheduling heuristic method based on the property.

Step 1:

Construct a list S of tardiness costs in which the tardiness costs are arranged in non-increasing order. Initially, $|S| = n$.

Step 2:

Arrange the ordered sequence from the last position of the last machine to the last position of the first machine and then the second last position of the first machine to the second position of last machine.

Step 3:

Calculate position r_j, $\sum_{i=1}^{k} D_{j[i]}$ and $\sum_{i=r_j+1}^{k} t_{j[i]}$ according to property 4.

Step 4:

For each machine j, order the jobs according to p_j/e_j in non-decreasing order.

Step 5:

Check the switched jobs to satisfy property 4 and change the r_j value if necessary.

Step 6:

Evaluate the objective function.

3.2.2 A Numerical Example

Now we use the numerical example from section 3.1.2 but with the job-independent earliness costs,tardiness costs and the due date costs. We now consider the problem of scheduling $n = 9$ jobs on $m = 3$ machines.

The tardiness costs are estimated as

$t_1 = 3, t_2 = 8, t_3 = 9, t_4 = 1, t_5 = 4, t_6 = 2, t_7 = 7, t_8 = 5, t_9 = 6.$

The earliness costs are estimated as

$e_1 = 2, e_2 = 1, e_3 = 1, e_4 = 1, e_5 = 1, e_6 = 3, e_7 = 3, e_8 = 1, e_9 = 2.$

And the due date costs are estimated as

$D_1 = 0, D_2 = 2, D_3 = 3, D_4 = 2, D_5 = 1, D_6 = 2, D_7 = 1, D_8 = 2, D_9 = 3.$

Then we order the tardiness costs sequence

$t_3 = 9, t_2 = 8, t_7 = 7, t_9 = 6, t_8 = 5, t_5 = 4, t_1 = 3, t_6 = 2, t_4 = 1.$

We assign jobs to machines according to step 1 and step 2. For each machine we have *number of job(corresponding tardiness cost)(due date cost)*:

$$\begin{array}{lll} \text{machine } 1: 4(1)(2) & 9(6)(3) & 7(7)(1) \\ \text{machine } 2: 6(2)(2) & 8(5)(2) & 2(8)(2) \\ \text{machine } 3: 1(3)(0) & 5(4)(1) & 3(9)(3). \end{array}$$

We then calculate r_j for each machine according to step 3.

$$r_1 = 2 \text{ of machine } 1 : t_7 = 7 \geq D_4 + D_9 + D_7 = 6$$

$$r_2 = 2 \text{ of machine } 2 : t_2 = 8 \geq D_6 + D_8 + D_2 = 6$$

$$r_3 = 2 \text{ of machine } 3 : t_3 = 9 \geq D_1 + D_5 + D_3 = 4.$$

Then $d_1^* = C_9 = 13, d_2^* = C_8 = 14, d_3^* = C_5 = 6$.

Now we calculate the value of the objective function

Figure 3: Job-dependent Heuristic Solution (A)

$f^*(d^*, s) = 3 \times 13 + 2 \times 13 + 1 \times 13 + 1 \times 9 + 7 \times 7 + 2 \times 14 + 2 \times 14 + 2 \times 14 + 8 \times 3 + 2 \times 8 + 6 \times 1 + 6 \times 3 + 5 \times 2 + 9 \times 3 = 321$.

We now apply step 4 to calculate the fractions.

$$\begin{array}{lll} \text{machine } 1: \frac{4}{1} & \frac{9}{2} & \frac{7}{3} \\ \text{machine } 2: \frac{6}{3} & \frac{8}{1} & \frac{2}{1} \\ \text{machine } 3: \frac{1}{2} & \frac{5}{1} & \frac{3}{1} \end{array}$$

We switch job 4 and job 9 on machine 1, job 6 and job 8 on machine 2, and place job 1 on the last position of machine 3, job 5 on the first position and job 3 on the second position. Then we obtain

$$\begin{array}{lll} \text{machine } 1: 9(6)(3) & 4(1)(2) & 7(7)(1) \\ \text{machine } 2: 8(5)(2) & 6(2)(2) & 2(8)(2) \\ \text{machine } 3: 5(4)(1) & 3(9)(3) & 1(3)(0). \end{array}$$

If we go back and check property 4, we find that machine 3 doesn't satisfy it if we still keep $r_3 = 2$. Then we adjust $r_3 = 1$ on machine 3 and obtain

$$\text{r=2 of machine } 1 : t_7 = 7 \geq D_9 + D_4 + D_7 = 6$$
$$\text{r=2 of machine } 2 : t_2 = 8 \geq D_8 + D_6 + D_2 = 6$$
$$\text{r=1 of machine } 3 : t_3 + t_1 = 12 \geq D_5 + D_3 + D_1 = 4.$$

Then $d_1^* = C_9 = 13, d_2^* = C_6 = 14, d_3^* = C_5 = 5$.

Figure 4: Job-dependent Heuristic Solution (B)

Now we calculate the optimal value of objective function
$f^*(d^*, s) = 3 \times 13 + 2 \times 13 + 1 \times 13 + 2 \times 4 + 7 \times 7 + 2 \times 14 + 2 \times 14 + 2 \times 14 + 6 \times 1 + 2 \times 8 + 5 \times 1 + 5 \times 3 + 3 \times 9 + 4 \times 3 = 300$.

4 Conclusion and Future Studies

We study job scheduling problems on both single machine and on independent identical parallel machines. The single machine problem considers that all the jobs are processed by only one machine. We introduced the job dependent CON problem and job independent CON problem on single machine. Then we considered the problem of scheduling n independent jobs on m parallel and identical machines. Given that each job has its own due date cost, earliness cost and tardiness cost, our objective is

to find the optimal due-date for each machine and the minimum costs. We proved the property 4 which can be applied to decide the optimal sequence by using the well known property of the slope and the intercept of a linear function. We also derived a scheduling heuristic method based on property 4 on parallel machines and property 2 on each single machine. Then we gave a numerical example which is used to show the above heuristic method clearly.

Additional extensions to the algorithm on parallel machine with job-dependent costs include nonlinear cost functions and set up different multiple machines.

A MATLAB Programm of Heuristic Solution

```
clc;
clear;

%We input the data we need such as number of machines and jobs,
%processing times and penalties
n=input('Please input numbers of jobs:');
m=input('Please input numbers of machines:');
for i=1:n
   t(i)=0;
end;
    fprintf('\n Please input the processing time of job
                in non-decreasing order:\n');
  for i=1:n
     fprintf('\n Please Input the processing time of job %d:',i);
     t(i)=input('');
  end;
p1=input('Please input the due date assignment cost per unit time P1:');
p2=input('Please input the earliness cost per unit time P2:');
p3=input('Please input the tardiness cost per unit time P3:');

%according to the algorithm, we calculate the value of k,r and lamda
k=ceil(n/m); r=ceil(k*(p3-p1)/(p2+p3)); lamda=zeros(m,k);
for i=1:r
 for j=1:m
    lamda(j,i)=k*p1+(i-1)*p2;
```

```
  end
end;
for i=r+1:k
 for j=1:m
    lamda(j,i)=(k+1-i)*p3;
 end
end;

%we show the position of each job
q=1; a=zeros(1,k*m);
for i=1:k
 for j=1:m
    a(q)=lamda(i,j);
    q=q+1;
 end
end;
s=zeros(m,k);
for i=1:k
 fprintf('\n Job %d should be processed in machine %d
 at position 2',i,i);
 s(i,2)=i;
end;
q=k+1;
for j=1:2:m
 for i=1:k
    if q<=n
       fprintf('\n Job %d should be processed in machine %d
```

```
       at position %d',q,i,j);
       s(i,j)=q;
       q=q+1;
    end
 end
end;

%we calculate the completion time and order them according
%to the non-decreasing order
c=zeros(m,k);
for i=1:k
 c(i,1)=t(s(i,1));
 for j=1:m-1
    if s(i,j+1)~=0
       c(i,j+1)=c(i,j)+t(s(i,j+1));
    end
 end
end;
r2=ceil(n*(p3-p1)/(p2+p3)); q=1; C=zeros(1,n);
for i=1:k
 for j=1:m
    if c(i,j)~=0
      C(s(i,j))=c(i,j);
      q=q+1;
    end
 end
end;
```

```
C1=zeros(1,n);
for i=1:n
 C1(i)=C(i);
end;
f=1; i=n-1;
while i>0 & f==1
 f=0;
 for j=1:i
   if C(j)>C(j+1)
       temp=C(j);
       C(j)=C(j+1);
       C(j+1)=temp;
       f=1;
   end;
 end
 i=i-1;
end;

%calculate the the earliness,tardiness and due date costs
dstar=C(r2); early=0; tardy=0;
for i=1:n
 if C(i)<dstar
    early=p2*(dstar-C(i))+early;
 else if C(i)>dstar
        tardy=p3*(C(i)-dstar)+tardy;
      else
         due=n*p1*dstar;
```

```
      end
   end
 end;

 %find the total costs according to the heuristic solution
 fprintf('\n The optimal due date is %3d',C(r2));
 fprintf('\n The total cost is %3d',early+tardy+due);

 %do the improved heuristic solution,find r of each machine and gamma
 rm=ceil(ceil(n/m)*p3/(p2+p3));
 gamma=zeros(1,k);
 for i=1:k
  for j=1:m
     if s(i,j)~=0
        if c(i,rm)<dstar
           gamma(i)=dstar-c(i,rm);
        end
     end
  end
 end;

%calculate the new completion time and order them
 for i=1:m
  for j=1:k
   if gamma(i)~=0
     c(i,j)=c(i,j)+gamma(i);
   end
```

```
 end
end;
f=1; i=n-1;
while i>0 & f==1
 f=0;
 for j=1:i
   if c(j)>c(j+1)
       temp=c(j);
       c(j)=c(j+1);
       c(j+1)=temp;
       f=1;
   end;
 end
 i=i-1;
 end;

 %find the new total costs and compare with the old one
 early2=0; tardy2=0; due2=0;
 for i=1:n
   if c(i)<dstar
     early2=p2*(dstar-c(i))+early2;
   else if c(i)>dstar
          tardy2=p3*(c(i)-dstar)+tardy2;
        else
          due2=n*p1*dstar;
        end
   end
```

```
end
fprintf('\n The total cost is %3d',early2+tardy2+due2);
```

After we run the above program in MATLAB, we can get the results as following:

```
 Please input numbers of jobs:9
 Please input numbers of machines:3

 Please input the processing time of job in non-decreasing order:
 Please input the processing time of job 1:1
 Please input the processing time of job 2:2
 Please input the processing time of job 3:3
 Please input the processing time of job 4:4
 Please input the processing time of job 5:5
 Please input the processing time of job 6:6
 Please input the processing time of job 7:7
 Please input the processing time of job 8:8
 Please input the processing time of job 9:9

 Please input the due date assignment cost per unit time P1:1
 Please input the earliness cost per unit time P2:2
 Please input the tardiness cost per unit time P3:3

 Job 1 should be processed in machine 1 at position 2
 Job 2 should be processed in machine 2 at position 2
 Job 3 should be processed in machine 3 at position 2
 Job 4 should be processed in machine 1 at position 1
 Job 5 should be processed in machine 2 at position 1
```

```
Job 6 should be processed in machine 3 at position 1
Job 7 should be processed in machine 1 at position 3
Job 8 should be processed in machine 2 at position 3
Job 9 should be processed in machine 3 at position 3

The optimal due date is 6
The total cost from the heuristic solution is 155
The cost from the improved heuristic solution is 15
```

REFERENCES

[1] Prabuddha De, Jay B. Ghosh,Charles E. Wells, 1993. Solving a generalized model for CON due date assignment and sequencing. *International journal of production economics*

[2] S.S. Panwalker and M.L. Smith, 1981. Common Due Date Assignment of Minimize Total Penalty for the One Machine Scheduling Problem. *Operation Research*

[3] George I. Adamopoulos and Costas P. Pappis, 1996. Single Machine Scheduling with Flow Allowance. *Journal of the Operational Research Society*

[4] M. A. Quaddus, 1987. A Generalized Model of Optimal Due-Date assignment by Linear Programming. *Journal of the Operational Research Society*

[5] T. C. E. Cheng, 1989. A Heuristic for Common Due-date Assignment and Job Scheduling on Parallel Machines. *Journal of the Operational Research Society*

[6] Kenneth R. Baker, Gary D. Scudder, 1989. Sequencing with Earliness and Tardiness Penalties: A Review. *Operation Research*

[7] Prabuddha De, Jay B. Ghosh,Charles E. Wells, 1991. On the Mutiple-machine Extension to a Common Due-date Assignment and Scheduling Prolem. *Journal of the Operational Research Society*