A STUDY OF GROUP PRODUCTION SCHEDULING

1979

TERUHIKO YOSHIDA

A STUDY OF GROUP PRODUCTION SCHEDULING
(生産のグループ・スケジューリングに関する研究)

1979

TERUHIKO YOSHIDA

CONTENTS

CHAPTER 1 GENERAL INTRODUCTION

1.1 Introduction

From the standpoint of production efficiency, the most desirable production type for manufacturing industries is that of mass production. However, quite a few manufacturing plants have adopted the small and medium lot size type of production. So far, efficient methods have been developed to increase productivity in small and medium lot size production. Group Technology (GT), which was developed by Mitrofanov, is one such useful method.

The concept of group technology has recently been introduced in the manufacturing areas of many forward looking companies. Group technology has been investigated from the standpoint of product design, commonality of tooling, and reduction of setup time. Intensive efforts have been made to establish effective classification and coding systems, which are fundamentals for group technology. In order to achieve additional benefits from group technology, this philosophy should be applied to other management areas, such as production scheduling. From this point of view, this thesis deals with the production scheduling problems associated with the concept of group technology, which will be referred to as "group production scheduling" or "group scheduling".

1.2 Group Technology

Group technology (often called "part-family manufacturing"), which was put forward by Mitrofanov,[1)] is one of the effective methods which aims at increasing the productivity of small and medium lot size manufacturing. Group technology is a technique to increase productivity by classifying a broad variety of parts having similarities as to shape, dimensions, and/or process route into several groups. By applying this technique to the small and medium lot size manufacturing, several advantages, such as mass production effect, possibility of flow-shop pattern on processing route, reduction of setup time and cost, and simplification of the material flow and handling, can be expected.

Group technology was first introduced into West Germany, then into Europe, later into U.S.A. and Japan. Now, group technology as a manufacturing concept has gained steady interest within the manufacturing industry all over the world.

The basis of group technology is a classification and coding system with which parts to be processed can be classified. Up to date, various classification and coding systems, most of which are based on the geometrical shape and/or processing routing, have been developed. The most representative one is the Aachen system in West Germany by Opitz.[2] Others are the Mitrofanov system (USSR), the VUOSO system (Czechoslovakia), the Brisch system (England), the TEKLA system (Norway), and the KK-1, 2 systems (Japan).[3] These systems have produced not a little effect in improving production efficiency, although they are not complete and universal. Furthermore, an additional significant saving from group technology is expected to be realized by applying the concept of group technology to other management areas. However, there still have been very few studies on this subject.[4~7]

1.3 Production scheduling

Scheduling is the allocation of jobs to be processed to a specific position on the time scale of a specified machine (or facility) in a workshop consisting of several machines. A job is a product or part to be processed and consists of a given sequence of operations. The processing of an operation requires the use of a particular machine for a given duration, the processing time of the operation. If attention is directed to a machine, there are several operations waiting for processing in a certain time span on that machine. Therefore, scheduling can be regarded as the problem of ordering the operations associated with each machine. In this sense, the term "sequencing" happens to be used synonymously with "scheduling." In general, it is defined that sequencing is concerned only with the ordering of operations on a single machine, while scheduling is a simultaneous and synchronized sequence on several machines. However, the two terms will be used interchangeably in this thesis, since such a usage of them seems to arouse no confusion.

Scheduling problems arise in a variety of situations. Within the range of scheduling problems, for example, there are problems of sequencing programs to be run at a computer center, and problems of sequencing patients at a hospital. The scheduling defined at the outset of this section is called "production scheduling," since it occurs in industrial production.

In general, there may be a number of schedules (sequences) in scheduling a given set of jobs on machines. Therefore, it is necessary to select one or several schedules from among them by a certain performance measure. This measure of performance is usually called "scheduling criterion" or simply "criterion", and many kinds of scheduling criteria are employed in production scheduling. The most representative ones include: total elapsed time (make-span), mean flow time, total tardiness, and facility utilization in the workshop.

A basic production scheduling problem is characterized by the following conditions:[8~10)]

(i) Jobs to be processed are available simultaneously for processing at time zero.

(ii) Each machine is continuously available for processing jobs.

(iii) Jobs consist of strictly-ordered sequences of operations.

(iv) The time required to complete a job consists of setup time and processing time, and is deterministic and known in advance.

(v) Each operation can be performed by only one machine.

(vi) There is only one machine of each type in the workshop.

(vii) Preemption is not allowed. (Once processing begins on a job, it is processed to completion without interruption.)

(viii) No overlapping. (The processing times of successive operations of a particular job may not be overlapped.)

(ix) Each machine can handle at most one operation at a time.

(x) Intermachine transportation times are ignored or included in processing times.

Several of these assumptions, of course, can frequently be relaxed and other assumptions can be added to these, which results in a different scheduling problem.

One of the most basic production scheduling problems is a single-stage (or single-machine) scheduling one, in which jobs, each of which consists of a single operation, are processed on a single machine. In the case of processing n jobs on a single machine, the total number of distinct schedules (sequences) to be evaluated is n!, which is the number of different permutations of n elements. In this sense, this problem is often called "job sequencing." For this problem, several useful scheduling rules for determining optimal schedules have been developed under various kinds of scheduling criteria.[8~12]

When the workshop consists of several machines, the shop is called a "flow shop" or "job shop" according to the type of flow pattern of jobs to be processed. A flow shop is one in which all the jobs pass identically from one machine to another. This type of flow pattern is typical for mass production. On the other hand, a job shop is one in which the flow of jobs is not unidirectional. This type of flow pattern occurs in small and medium lot size production.

Flow-shop scheduling problems are complicated as compared with single-machine scheduling ones, and hence only a few theoretical results have been reported. A well-known and practical one is Johnson's theorem[13] for the two-machine flow-shop scheduling problem. For a more than two-machine flow-shop scheduling problem, in general, no theorem which gives easily an optimal schedule has been developed. It may be necessary to resort to general purpose methodologies, such as a dynamic programming approach, a branch-and-bound method, or a heuristic procedure, all of which can be applied to solve complicated combinatorial problems including scheduling problems.

Job-shop scheduling problems are much more complicated and not yet completely solved. The only case which is theoretically solved is the two-job, m-machine job-shop scheduling problem.[14)] In principle, it may be possible that there are $(n!)^m$ alternatives when n jobs are to be processed on m machines in a job shop. It is possible theoretically to obtain an optimal schedule by enumerating all possible schedules and selecting a schedule according to a certain measure of performance. This, however, is not practical, because it requires substantial computational efforts, particularly when the number of jobs is large. For example, there exist approximately 1.4×10^7 possible schedules even in the case of $m = n = 6$. Several attempts have been made to solve the job-shop scheduling problems by applying the general purpose methodologies. Regrettably, no successful result has been reported.

The scheduling models mentioned above are static ones, since jobs are available simultaneously for processing. On the other hand, in a dynamic job-shop scheduling model where jobs arrive at random over a certain time, it is almost impossible to analytically determine an optimal schedule. For this type of model, an effective approach is a scheduling simulation. In the job shop, jobs to be processed on a specified machine make a queue in front of that machine, so that the shop behaves like a network of queues. In this case, the processing order of jobs is determined by means of dispatching decisions. The study of scheduling in a dynamic job shop has made considerable progress with the use of computer simulation models. With the aid of these models, a large number of scheduling rules (or dispatching rules) for giving priorities to jobs in the queue in front of each of the machines have been developed, and broad conjectures about scheduling procedures have been obtained.[8,9]

1.4 Group production scheduling

As stated in the previous sections, it is expected that, in addition to the benefits from the pure-production technological viewpoints attained by group technology, an additional benefit is achieved by applying this philosophy to the production scheduling. In processing a large variety of jobs, several results, such as reduction of setup time, learning effects, and reduction of fraction defective, will be obtained by processing the jobs with the same or similar operations in succession.

Based on the above consideration, production scheduling models of a new type have been developed for the purpose of improving the productivity in small and medium lot size manufacturing. In the scheduling models, jobs having the same or similar operations are assumed to be classified into the same group and processed in succession. In this thesis, the production scheduling which is associated with the concept of group technology is referred to as "group production scheduling" or "group scheduling" for short.

First, group scheduling models under static conditions are considered. The fundamental assumptions of the models are as follows:

(i) Jobs to be processed are classified into several groups and jobs within the same group are processed in succession.

(ii) Group processing time required for completion of a group consists of group setup time and the sum of job processing times contained in each group.

(iii) Group setup time necessary to process a group is independent of the sequence of groups.

(iv) Job setup time needed to process a job is independent of the sequences of groups and jobs, and is included in the job processing time.

(v) In the case of multiple production stages, all jobs and groups are processed in a flow-shop pattern.(All jobs and groups follow the same path from one stage to another.) Furthermore, the ordering of groups and jobs is assumed to be the same on each machine.(No passing of groups and jobs is allowed.)

Table 1.1 shows the group scheduling where jobs are classified into N groups, each of which consists of n_i jobs ($i=1, 2, \ldots, N$).

Table 1.1 Group scheduling under static conditions

Group	G_1				G_2				...	G_N			
Job	J_{11}	J_{12}	...	J_{1n_1}	J_{21}	J_{22}	...	J_{2n_2}	...	J_{N1}	J_{N2}	...	J_{Nn_N}

In the group scheduling models defined above, there are $N! \times \prod_{i=1}^{N} n_i!$ feasible schedules on each machine. On the other hand, in many conventional scheduling problems where there exists no precedence relation among jobs, the number of sequences to be evaluated is n! on each machine in the case of n jobs. Therefore, the conventional scheduling can be regarded as a kind of group scheduling in which only one group consisting of n jobs is involved.

In the group scheduling problem, optimal decisions are to be made as to the sequence of groups classified and the sequence of jobs in each group. In this thesis, they are called "group sequence" and "job sequence," respectively. Furthermore, a schedule in which both group and job sequences are specified is called a "group schedule."

There is a string problem which seems to be similar to the group scheduling problem. However, significant differences exist between both problems in that the string problem has no background of group technology, considers only the sequence of groups classified excluding the sequence of jobs in each group, and does not include group setup times in its model.

In the string problem, theoretical analyses have been made as to the single-stage problems of minimizing certain measures of performance[15,16] and the two-stage problems of minimizing the total elapsed time.[17,18] However, there have been only a few studies treating both sequences of groups and jobs simultaneously.[19,20] In the first part of this thesis, analyses of the group scheduling problems under static conditions are performed and several effective theorems and algorithms for determining the optimal group schedules are developed.

In real situations in manufacturing plants, jobs arrive at the shop randomly over time. Hence, it is important to consider the scheduling problem under dynamic conditions. In the latter part of the thesis, the group scheduling problem under dynamic conditions is also considered under the following fundamental conditions:

(i) Jobs to be processed are classified into several setup groups.

(ii) Group setup times are dependent on the sequence of the groups to which jobs belong.

1.5 Outline of thesis

In Chapters 2 to 6, group scheduling models under static conditions are dealt with. Chapter 2 introduces a basic single-stage group scheduling model. First, theorems are offered to obtain the optimal group schedules under two kinds of criteria — the minimum mean flow time and the minimum weighted mean flow time. Second, an algorithm for determining a group schedule minimizing the total tardiness is developed.

In Chapter 3, a group scheduling model with sequence-dependent setup times is developed. In this model, group setup times are assumed to be dependent on the sequence of groups. It is shown that the group scheduling problem of minimizing the total elapsed time can be reduced to a traveling salesman problem. Under the criteria of the minimum mean flow time and the minimum total tardiness, optimizing algorithms are developed to determine optimal group schedules, respectively.

In Chapter 4, a theoretical analysis is made to the group scheduling on the multiple production stages under the criterion of the minimum total elapsed time. First, the well-known Johnson's theorem is extended to the two-stage scheduling problem with setup times separated. Second, the optimal group schedules on the multiple production stages are determined for those special cases where group setup times and job processing times have well-defined relationships to one another.

In Chapter 5, a branch-and-bound method is applied to solve multistage group scheduling problems. The branching procedure for group scheduling is clarified, and lower bounds for the total elapsed time and the weighted mean flow time are developed. Optimizing algorithms are presented to find optimal group schedules, and the effectivenesses of the algorithms are examined with numerical experiments.

Chapter 6 develops a multistage group scheduling model with variable processing times and costs depending on machining conditions. The optimal group schedule minimizing the total elapsed time with the minimum number of tardy jobs is determined, and then the optimal machining speeds minimizing the total production cost are decided under the determined optimal group schedule.

Chapter 7 deals with a dynamic group scheduling model where jobs arrive at the workshop randomly over time. Scheduling simulations are run to investigate the effect of types of flow patterns — job-shop, near-flow-shop, and flow-shop patterns on the flow time performances. In addition, the effect of the setup time ratio, defined as the ratio of the mean setup time to the mean processing time, on the performances for several scheduling rules is studied in the experiments.

CHAPTER 2 GROUP SCHEDULING ON A SINGLE PRODUCTION STAGE

2.1 Introduction

A single-stage (or single-machine) scheduling problem is the most basic one in production scheduling. In this problem the number of schedules to be evaluated is n! when n jobs are to be processed on a single machine. Hence, one can find an optimal schedule by paying attention to the permutations of job indices. Thus, this problem is called "job sequencing." This problem has been studied by many researchers and lots of theorems and algorithms which give optimal schedules have been proposed and developed under various kinds of criteria.[1~4]

Even for group scheduling, a single-stage scheduling problem is a fundamental in the study of sequencing and scheduling. This chapter deals with the group scheduling on a single production stage and develops the optimal group scheduling under three kinds of criteria.[5,6]

The scheduling criteria employed in this chapter are the following:

(i) The minimum mean flow time

(ii) The minimum weighted mean flow time

(iii) The minimum total tardiness

The problem is to determine the optimal group schedule (optimal sequences of groups and jobs) minimizing each of the above performance measures.

It is supposed that jobs to be processed are classified into N groups, each of which consists of n_i jobs ($i = 1, 2, \ldots, N$). Let $J_{i\xi}$ ($i = 1, 2, \ldots, N$, $\xi = 1, 2, \ldots, n_i$) denote the ξth job in group G_i ($i = 1, 2, \ldots, N$) and $p_{i\xi}$ and $d_{i\xi}$ ($i = 1, 2, \ldots, N$, $\xi = 1, 2, \ldots, n_i$) denote the job processing time including job setup time and the due date of job $J_{i\xi}$, respectively. Furthermore, let S_i ($i = 1, 2, \ldots, N$) denote the group setup time of group G_i. The symbol (i) is used to signify a job or a group sequenced in the ith position for a group schedule.

In the scheduling problem defined, the total elapsed time, which is a fundamental and important criterion in production scheduling, is no longer a suitable one. The total elapsed time is given by

$$F_{max} = \sum_{i=1}^{N} S_{(i)} + \sum_{i=1}^{N}\sum_{\xi=1}^{n_i} p_{(i)(\xi)} \tag{2.1}$$

Hence, the total elapsed time is not dependent on the order of groups and jobs for a single-stage problem.

2.2 Minimizing mean flow time and weighted mean flow time

The criteria of the minimum mean flow time and the minimum weighted mean flow time are tractable even in the group scheduling. In this section, a theorem which gives the optimal group schedule minimizing the mean flow time is offered, and then it is extended to the criterion of the minimum weighted mean flow time.

The completion time of $J_{i\xi}$, the ξth job in the ith group is

$$C_{(i)(\xi)} = \sum_{j=1}^{i-1} (S_{(j)} + P_{(j)}) + S_{(i)} + \sum_{\nu=1}^{\xi} p_{(i)(\nu)} \tag{2.2}$$

where $P_{(j)} (= \sum_{\xi=1}^{n_j} p_{(j)(\xi)})$ is the total processing time of $G_{(j)}$.
Since the ready times of all jobs are to be zero, the flow time of $J_{(i)(\xi)}$ is simply

$$F_{(i)(\xi)} = C_{(i)(\xi)} \tag{2.3}$$

Thus, the mean flow time is obtained by

$$\begin{aligned} \bar{F} &= \sum_{i=1}^{N}\sum_{\xi=1}^{n_i} F_{(i)(\xi)} \Big/ \sum_{i=1}^{N} n_i \\ &= \frac{1}{M}\sum_{i=1}^{N} n_{(i)} \sum_{j=1}^{i-1} Q_{(j)} + \frac{1}{M}\sum_{i=1}^{N} n_{(i)} S_{(i)} + \frac{1}{M}\sum_{i=1}^{N}\sum_{\xi=1}^{n_i}\sum_{\nu=1}^{\xi} p_{(i)(\nu)} \end{aligned} \tag{2.4}$$

where $Q_{(i)}$ $(= S_{(i)} + P_{(i)})$ is the group processing time of $G_{(i)}$ and M $(= \sum_{i=1}^{N} n_i)$ is the number of all jobs.

In the above equation, the second term is a constant. The first term is concerned with the group sequence and is independent of the job sequences because $Q_{(j)}$ is a constant. The last term is concerned with the job sequence in each group, and is not influenced by the group sequence. Hence, the group sequence and the job sequences can be determined independently of each other.

The first term is minimized by ordering the groups in nondecreasing order of Q_i/n_i. The last one is minimized by ordering the jobs in nondecreasing order of job processing times for each group. Sequencing the jobs in nondecreasing order of processing times is usually called "shortest-processing-time (SPT) sequencing" and is the most important concept in the entire subject of scheduling. It performs with surprising efficacy even in dynamic job shop scheduling, as will be shown in the last chapter of this thesis.

Thus the results are stated formally as a theorem in the following way.

Theorem 2.1 In a single-stage group scheduling problem, the mean flow time is minimized by ordering the jobs in each group and the groups, respectively, such that

$$P_{(i)(1)} \le P_{(i)(2)} \le \cdots \le P_{(i)(n_i)} \quad (i=1,\ 2,\ldots,\ N)$$

and

$$\frac{S_{(1)}+P_{(1)}}{n_{(1)}} \le \frac{S_{(2)}+P_{(2)}}{n_{(2)}} \le \cdots \le \frac{S_{(N)}+P_{(N)}}{n_{(N)}}$$

In some cases, jobs do not have equal importance. A value or weighting factor, $w_{i\xi}(i=1,\ 2,\ldots,\ N,\ \xi=1,\ 2,\ \ldots,\ n_i)$ is assumed to be given for each job to describe its relative importance.

The weighted mean flow time is given by the following equation, similar to equation (2.4).

$$\bar{F}_w = \frac{1}{M}\sum_{i=1}^{N}\sum_{\xi=1}^{n_i} w_{(i)(\xi)}\sum_{j=1}^{i-1}Q_{(j)} + \frac{1}{M}\sum_{i=1}^{N}\sum_{\xi=1}^{n_i} w_{(i)(\xi)}S_{(i)} + \frac{1}{M}\sum_{i=1}^{N}\sum_{\xi=1}^{n_i}\sum_{\nu=1}^{\xi} w_{(i)(\nu)}P_{(i)(\nu)} \qquad (2.5)$$

In the case of minimizing the weighted mean flow time, the following theorem, which is an extension of Theorem 2.1, holds.

Theorem 2.2 In a single-stage group scheduling problem, the weighted mean flow time is minimized by ordering the jobs in each group and the groups, respectively, such that

$$\frac{p_{(i)(1)}}{w_{(i)(1)}} \le \frac{p_{(i)(2)}}{w_{(i)(2)}} \le \cdots \le \frac{p_{(i)(n_i)}}{w_{(i)(n_i)}} \quad (i=1,\ 2,\ldots,\ N)$$

and

$$\frac{S_{(1)} + P_{(1)}}{\sum_{\xi=1}^{n_1} w_{(1)(\xi)}} \le \frac{S_{(2)} + P_{(2)}}{\sum_{\xi=1}^{n_2} w_{(2)(\xi)}} \le \cdots \le \frac{S_{(N)} + P_{(N)}}{\sum_{\xi=1}^{n_N} w_{(N)(\xi)}}$$

The proof of this theorem is omitted since it is proved in much the same way as in the case of minimizing the mean flow time.

2.3 Minimizing total tardiness

The criterion concerning jobs' due dates, especially the minimum total tardiness, is important in production scheduling. Not a few efforts have been made to solve the conventional scheduling problem of minimizing the total tardiness. However, none of the complete solution procedures was presented because of the complexities of the problem. A heuristic algorithm for determining a suboptimal schedule has been proposed by Wilkerson and Irwin.[7] For determining an optimal schedule, Emmons[8] gave several theorems which establish the relative order of pairs of jobs and proposed an efficient implicit enumeration algorithm.

In this section, an extension of Emmons' theory to the group scheduling is made. Several conditions under which certain groups precede others in an optimal group schedule are offered to find the optimal sequences of groups and jobs. As to the job sequence, two cases are considered: (i) job sequences are predetermined, (ii) job sequences are not predetermined. Then, efficient algorithms for determining the optimal group schedule or the near optimal group schedule under both conditions are proposed. An optimal group schedule can be obtained by evaluating a few schedules for moderate size problems. The algorithms eliminate most sequences from consideration.

2.3.1 Total tardiness

Since the completion time of $J_{(i)(\xi)}$, the ξth job in the ith group, is given by equation (2.2), the tardiness of the job is

$$T_{(i)(\xi)} = \max\ (C_{(i)(\xi)} - d_{(i)(\xi)},\ 0) \tag{2.6}$$

Then, the total tardiness of all jobs included in $G_{(i)}$ is

$$T_{(i)} = \sum_{\xi=1}^{n_i} T_{(i)(\xi)} = \sum_{\xi=1}^{n_i} \max\ (C_{(i)(\xi)} - d_{(i)(\xi)},\ 0) \tag{2.7}$$

Hence, from equations (2.2) and (2.7) the total tardiness of all jobs in all groups is

$$T = \sum_{i=1}^{N} \sum_{\xi=1}^{n_i} \max\ (\sum_{j=1}^{i-1} Q_{(j)} + S_{(i)} + \sum_{\nu=1}^{\xi} p_{(i)(\nu)} - d_{(i)(\xi)},\ 0) \tag{2.8}$$

Hereafter, groups will be indexed in order of nondecreasing group processing time Q_j including group setup time S_j.

Let A_i and B_i ($i = 1, 2, \ldots, N$) be the sets of indices of groups that, at any point, have been shown to come after and before G_i in an optimal group schedule, respectively. Furthermore, let Z_i ($i = 1, 2, \ldots, N$) be the set of indices of all jobs in G_i.

2.3.2 Scheduling algorithm I (In the case of predetermined job sequence)

As to group scheduling for the minimum-total-tardiness criterion, we obtain the following theorems and corollaries that establish the relative order of pairs of groups in an optimal group schedule. These are extensions of Emmons' theorems and corollaries.

Theorem 2.3 For any two groups G_i and G_j with $Q_i \le Q_j$, if the following conditions are satisfied, then G_i precedes G_j in an optimal group schedule.

(I) In the case of $n_i \ge n_j$, n_j inequalities (2.9) hold for n_j sets of two jobs $J_{i\xi}$ and $J_{j\eta}$ ($\xi \in Z_i$, $\eta \in Z_j$).

$$d_{i\xi} + \sum_{\nu=\xi+1}^{n_i} p_{i\nu} \le \max\left(\sum_{k \in B_j} Q_k + Q_j,\ d_{j\eta} + \sum_{\nu=\eta+1}^{n_j} p_{j\nu}\right) \tag{2.9}$$

(II) In the case of $n_i < n_j$, in addition to the above n_i inequalities, $(n_j - n_i)$ inequalities (2.10) hold for the remaining $(n_j - n_i)$ jobs $J_{j\eta}$ ($\eta \in Z_j$).

$$d_{j\eta} + \sum_{\nu=\eta+1}^{n_j} p_{j\nu} \ge Q - \sum_{k \in A_i} Q_k \tag{2.10}$$

where $Q = \sum_{i=1}^{N} Q_i$.

Proof. Let S be any schedule in which G_j precedes G_i. Consider a schedule S' that differs from S only in that G_j and G_i are interchanged. We shall show that interchanging the two groups must decrease, or possibly leave unchanged, the total tardiness. Denote by X and Y the times at which G_j begins and G_i ends, respectively, in S (see Fig.2.1). Clearly, all groups

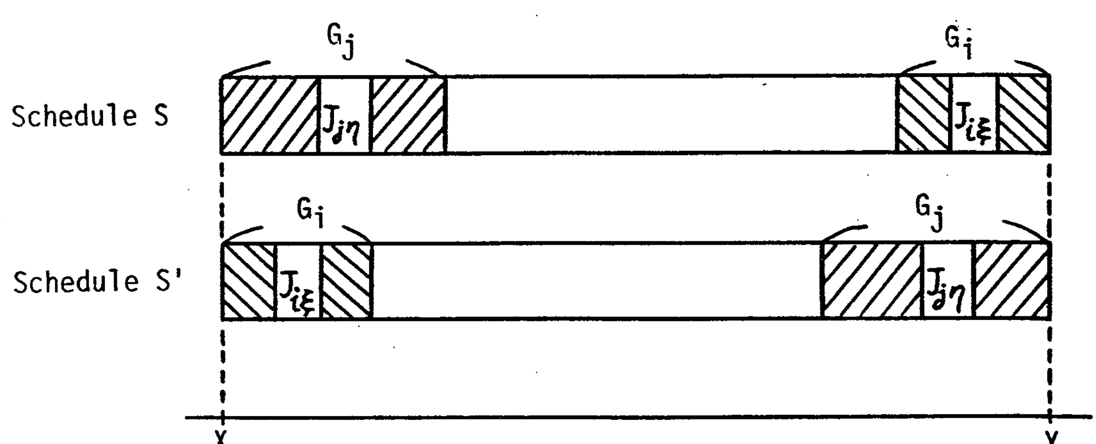


Fig. 2.1 The effect of interchanging two groups

that precede G_j or follow G_i in S are unaffected. Hence, the tardinesses of those groups remain unchanged. All groups between G_j and G_i are advanced in time by the amount of $(Q_j - Q_i) \geq 0$, which can only decrease or leave unchanged their tardiness. Let ΔT_i^- and ΔT_j^+ be the decrease of tardiness of G_i and the increase of tardiness of G_j, respectively. Hence, for the proof of Theorem 2.3, it is sufficient to show $\Delta T_i^- \geq \Delta T_j^+$ under condition (I) or (II).

Let $\Delta T_{i\xi}^-$ and $\Delta T_{j\eta}^+$ be the decrease of tardiness of $J_{i\xi}$ and the increase of tardiness of $J_{j\eta}$, respectively, for which inequality (2.9) holds. Then, $\Delta T_{i\xi}^- \geq \Delta T_{j\eta}^+$ is shown as follows.

Inequality (2.9) implies that either $d_{i\xi} + \sum_{\nu=\xi+1}^{n_i} p_{i\nu} \leq \sum_{k\in B_j} Q_k + Q_j$ or $d_{i\xi} + \sum_{\nu=\xi+1}^{n_i} p_{i\nu} \leq d_{j\eta} + \sum_{\nu=\eta+1}^{n_j} p_{j\nu}$ (or both).

Case (a). Suppose $d_{i\xi} + \sum_{\nu=\xi+1}^{n_i} p_{i\nu} \leq \sum_{k\in B_j} Q_k + Q_j$. From the meaning of B_j, $X \geq \sum_{k\in B_j} Q_k$, so that $d_{i\xi} + \sum_{\nu=\xi+1}^{n_i} p_{i\nu} \leq X + Q_j \leq Y$. Hence, $\Delta T_{i\xi}^- = Y - \sum_{\nu=\xi+1}^{n_i} p_{i\nu} - \max(X + S_i + \sum_{\nu=1}^{\xi} p_{i\nu}, d_{i\xi}) = Y - \max(X + Q_i, d_{i\xi} + \sum_{\nu=\xi+1}^{n_i} p_{i\nu})$.

On the other hand, for $\Delta T_{j\eta}^+$,

$$\Delta T_{j\eta}^+ = \begin{cases} 0, \text{ if } d_{j\eta} \geq Y - \sum_{\nu=\eta+1}^{n_j} p_{j\nu} \\ Y - \max(X + Q_j, d_{j\eta} + \sum_{\nu=\eta+1}^{n_j} p_{j\nu}), \text{ otherwise} \end{cases}$$

If $d_{j\eta} \geq Y - \sum_{\nu=\eta+1}^{n_j} p_{j\nu}$, then $\Delta T_{i\xi}^- - \Delta T_{j\eta}^+ = \Delta T_{i\xi}^- \geq 0$

If $d_{j\eta} < Y - \sum_{\nu=\eta+1}^{n_j} p_{j\nu}$, then $\Delta T_{i\xi}^- - \Delta T_{j\eta}^+ = \max(X + Q_j, d_{j\eta} + \sum_{\nu=\eta+1}^{n_j} p_{j\nu})$ $- \max(X + Q_i, d_{i\xi} + \sum_{\nu=\xi+1}^{n_i} p_{i\nu}) \geq 0$, since $X + Q_j \geq \max(X + Q_i, d_{i\xi} + \sum_{\nu=\xi+1}^{n_i} p_{i\nu})$.

Case (b). Suppose $d_{i\xi} + \sum_{\nu=\xi+1}^{n_i} p_{i\nu} \leq d_{j\eta} + \sum_{\nu=\eta+1}^{n_j} p_{j\nu}$. We consider three cases.

(i) If $d_{i\xi} + \sum_{\nu=\xi+1}^{n_i} p_{i\nu} \leq d_{j\eta} + \sum_{\nu=\eta+1}^{n_j} p_{j\nu} < Y$, then $\Delta T_{i\xi}^- = Y - \max(X + Q_i, d_{i\xi} + \sum_{\nu=\xi+1}^{n_i} p_{i\nu})$, $\Delta T_{j\eta}^+ = Y - \max(X + Q_j, d_{j\eta} + \sum_{\nu=\eta+1}^{n_j} p_{j\nu})$.

Hence, $\Delta T_{i\xi}^- - \Delta T_{j\eta}^+ = \max(X + Q_j, d_{j\eta} + \sum_{\nu=\eta+1}^{n_j} p_{j\nu}) - \max(X + Q_i, d_{i\xi} + \sum_{\nu=\xi+1}^{n_i} p_{i\xi}) \geq 0$, since $X + Q_j \geq X + Q_i$ and $d_{j\eta} + \sum_{\nu=\eta+1}^{n_j} p_{j\nu} \geq d_{i\xi} + \sum_{\nu=\xi+1}^{n_i} p_{i\nu}$.

(ii) If $d_{i\xi} + \sum_{\nu=\xi+1}^{n_i} p_{i\nu} \leq Y \leq d_{j\eta} + \sum_{\nu=\eta+1}^{n_j} p_{j\nu}$, then $\Delta T_{i\xi}^- = Y - \max(X + Q_i, d_{i\xi} + \sum_{\nu=\xi+1}^{n_i} p_{i\nu})$, $\Delta T_{j\eta}^+ = 0$.

Hence, $\Delta T_{i\xi}^- - \Delta T_{j\eta}^+ = \Delta T_{i\xi}^- \geq 0$.

(iii) If $Y < d_{i\xi} + \sum_{\nu=\xi+1}^{n_i} p_{i\nu} \leq d_{j\eta} + \sum_{\nu=\eta+1}^{n_j} p_{j\nu}$, then $\Delta T_{i\xi}^- = \Delta T_{j\eta}^+ = 0$.

Thus in all cases, $\Delta T^-_{i\xi} \geq \Delta T^+_{j\eta}$ for two jobs $J_{i\xi}$ and $J_{j\eta}$ satisfying inequality (2.9). Then $T^-_i \geq \Delta T^+_j$ is shown as follows.

(A) $n_i \geq n_j$: From condition (I), $\Delta T^-_{i\xi} \geq \Delta T^+_{j\eta}$ $(\xi \in Z_i, \eta \in Z_j)$ for n_j sets of two jobs $J_{i\xi}$ and $J_{j\eta}$ $(\xi \in Z_i, \eta \in Z_j)$ and $\Delta T^-_{i\xi} \geq 0$ $(\xi \in Z_i)$ for the remaining $(n_i - n_j)$ jobs $J_{i\xi}$ $(\xi \in Z_i)$. Hence, $\Delta T^-_i \geq T^+_j$.

(B) $n_i < n_j$: From condition (II), $\Delta T^-_{i\xi} \geq \Delta T^+_{j\eta}$ $(\xi \in Z_i, \eta \in Z_j)$ for n_i sets of two jobs $J_{i\xi}$ and $J_{j\eta}$ $(\xi \in Z_i, \eta \in Z_j)$ and $\Delta T^+_{j\eta} \geq 0$ $(\eta \in Z_j)$ for the remaining $(n_j - n_i)$ jobs $J_{j\eta}$ $(\eta \in Z_j)$. Hence, $\Delta T^-_i \geq \Delta T^+_j$.

(Q. E. D.)

Corollary 2.1 If Theorem 2.3 is satisfied for G_1 and G_i $(\forall i > 1)$, where G_1 has the least group processing time, then G_1 is the first group in an optimal group schedule.

Corollary 2.2 If Theorem 2.3 is satisfied for G_N and G_i $(\forall i < N)$, where G_N has the largest group processing time, then G_N is the last group in an optimal group schedule.

Theorem 2.4 For any two groups G_i and G_j with $Q_i \leq Q_j$, if the following conditions are satisfied, then G_j precedes G_i in an optimal group schedule.

(I) In the case of $n_i \geq n_j$, n_j sets of inequalities (2.11) and (2.12) hold for n_j sets of two jobs $J_{i\xi}$ and $J_{j\eta}$ $(\xi \in Z_i, \eta \in Z_j)$, and $(n_i - n_j)$ inequalities (2.13) hold for the remaining jobs $J_{i\xi}$ $(\xi \in Z_i)$.

$$d_{i\xi} + \sum_{\nu=\xi+1}^{n_i} p_{i\nu} > \max\Big(\sum_{k \in B_j} Q_k + Q_j,\ d_{j\eta} + \sum_{\nu=\eta+1}^{n_j} p_{j\nu}\Big) \tag{2.11}$$

$$d_{i\xi} + \sum_{\nu=\xi+1}^{n_i} p_{i\nu} + Q_i \geq Q - \sum_{k \in A_j} Q_k \tag{2.12}$$

$$d_{i\xi} + \sum_{\nu=\xi+1}^{n_i} p_{i\nu} \geq Q - \sum_{k \in A_j} Q_k \tag{2.13}$$

(II) In the case of $n_i < n_j$, n_j sets of inequalities (2.11) and (2,12) are satisfied for n_i sets of two jobs $J_{i\xi}$ and $J_{j\eta}$ $(\xi \in Z_i, \eta \in Z_j)$.

Since this theorem can be proved in much the same way as the proof of Theorem 2.3, we shall only give a brief outline of the proof.

Let S be any schedule in which G_i precedes G_j and S' be a schedule in which G_i is immediately after G_j, as shown in Fig. 2.2. It can be shown in

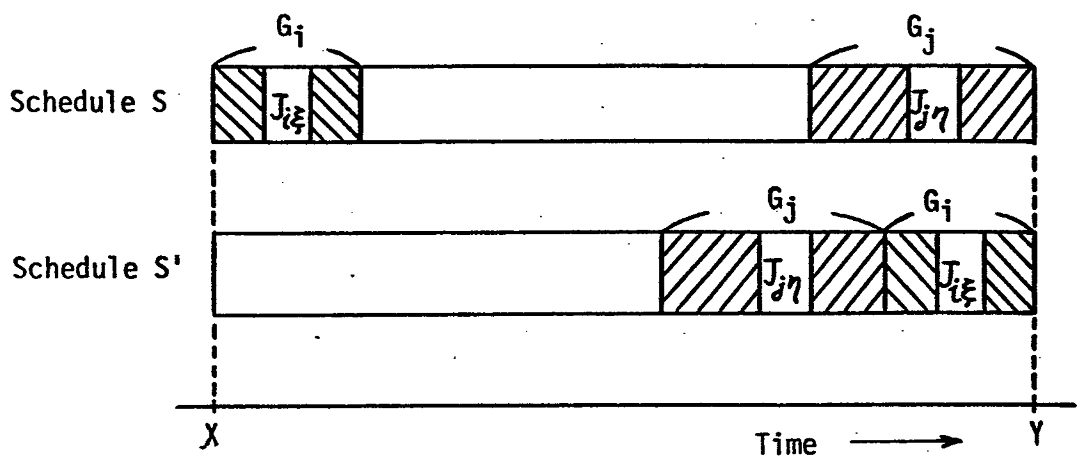


Fig. 2.2 The effect of postponing a group

the same way as before that $\Delta T^-_{j\eta} \geq \Delta T^+_{i\xi}$ for a set of two jobs $J_{j\eta}$ and $J_{i\xi}$ satisfying (2.11) and (2.12) and then $\Delta T^-_j \geq \Delta T^+_i$.

Corollary 2.3 If Theorem 2.3 is satisfied for G_s having the least number of jobs and G_i such that $Q_i \leq Q_s$ ($\forall i < s$) and Theorem 2.4 holds for G_s and G_j such that $Q_s \leq Q_j$ ($\forall j > s$), then G_s is the last group in an optimal group schedule.

Theorem 2.5 For any two groups G_i and G_j with $Q_i \leq Q_j$, if n_j inequalities (2.14) hold for $J_{j\eta}$ ($\eta = 1, 2, \ldots, n_j$), then G_i precedes G_j in an optimal group schedule.

$$d_{j\eta} + \sum_{\nu=\eta+1}^{n_j} p_{j\nu} \geq Q - \sum_{k \in A_i} Q_k \qquad (2.14)$$

Proof. Let us consider schedules S and S' similar to ones shown in Fig.2.1. Select $J_{j\eta}$ which satisfies (2,14), then let $\Delta T^+_{j\eta}$ be the increase of tardiness of this job. Then $d_{j\eta} + \sum_{\nu=\eta+1}^{n_j} p_{j\nu} \geq Q - \sum_{k\in A_i} Q_k \geq Y$. Hence, the due date of $J_{j\eta}$ is $d_{j\eta} \geq Y - \sum_{\nu=\eta+1}^{n_j} p_{j\nu}$. Therefore, $\Delta T_{j\eta}^{+} = 0$. If the conditions of Theorem 2.5 hold, then $\Delta T_{j\eta}^{+} = 0$ for all jobs in G_j. Hence, $\Delta T_j^{+} = 0$. On the other hand, $\Delta T_i^{-} \geq 0$ for G_i. Therefore, the total tardiness under S' is less than the total tardiness under S.

(Q. E. D.)

The above theorems and corollaries are effective to determine the optimal group sequence because we can eliminate most sequences from consideration by using the sufficient conditions.

By using the above theorems and corollaries and branching whenever necessary, the following algorithm is proposed to find an optimal group sequence. For large problems where a good deal of branching can be expected, the standard branch-and-bound technique which will be explained in later chapters should be incorporated into this algorithm.

< Algorithm I for determining an optimal group schedule >

Step 1. If the due dates of all jobs in a group are later than the total processing time of the remaining groups, then place this group last. Repeat as often as possible. If the set of the remaining groups, G, is empty, then terminate. Otherwise, go to Step 2.

Step 2. If Corollary 2.2 can be used for the group having the largest group processing time, place this group last. Repeat as often as possible, and go back to Step 1. If $G = \phi$, then terminate. Otherwise, go to Step 3.

Step 3. If Corollary 2.3 can be used for the group having the least number of jobs, place this group last. Repeat as often as possible, and go back to Step 1. If $G = \phi$, then terminate. Otherwise, go to Step 4.

Step 4. If $G = \phi$, then compute the total tardiness for the relevant node and go to Step 9. Otherwise, go to Step 5.

Step 5. Generate D, the set of candidates of groups that can be first in an optimal schedule in the following way:

(i) Let D = G and E = G.

(ii) Let the first group in E be G_x.

(iii) Eliminate from D all groups G_k $(k > x)$ which are shown to come after G_x by using Theorem 2.3 or Theorem 2.5 and remove G_x from E.

(iv) If $E \neq \phi$, then go back to (ii).

Step 6. If D contains only one group G_y, place G_y first and subtract the group processing time Q_y from all due dates in the remaining groups and go back to Step 4. Otherwise, go to Step 7.

Step 7. If Theorem 2.4 holds for the last group G_m in D and any other group G_i $(i < m)$, then place G_m first and subtract the group processing time Q_m from all due dates in the remaining groups and go back to Step 4. Otherwise, go to Step 8.

Step 8. Branch on the assumptions that (i) G_{m-1} precedes G_m and that (ii) G_m precedes G_{m-1}. For case (i), remove G_m from D, and for case (ii) remove G_{m-1} from D. Go back to Step 6.

Step 9. Select a node and denote the remaining groups for this node by G and go back to Step 5. If there is no remaining node, then go to Step 10.

Step 10. Find a schedule having the minimum total tardiness among the schedules for the nodes derived. This is the optimal group sequence.

2.3.3 Scheduling algorithm II (In the case that job sequences are not predetermined)

Since group sequence and job sequences depend on each other, determining the optimal group schedule is more difficult than in the previous case. In this case the following theorem holds.

Theorem 2.6 For any two jobs $J_{i\xi}$ and $J_{i\xi+1}$ with $p_{i\xi} \leq p_{i\xi+1}$ $(\xi = 1, 2, \ldots,$

n_i-1), if $d_{i\xi} \leq \max(p_{i\xi+1}, d_{i\xi+1})$, then the optimal job sequence of G_i is the SPT schedule regardless of the group start time t ($\gtrsim 0$), the time when the group begins to be processed.

Proof. Let the group start time of G_i be time zero. Then the revised due dates are $d_{i\xi} - t$ ($\xi = 1, 2, \ldots, n_i$).

By hypothesis, (1) $d_{i\xi} - t \leq d_{i\xi} \leq p_{i\xi+1}$ $(d_{i\xi} \leq p_{i\xi+1})$

(2) $d_{i\xi} - t \leq d_{i\xi+1} - t$ $(d_{i\xi} \leq d_{i\xi+1})$.

Hence, $d_{i\xi} - t \leq \max(p_{i\xi+1}, d_{i\xi+1} - t)$ ($\xi = 1, 2, \ldots, n_i-1$).

Supposing $n_i = n_j = 1$ and replacing G_i and G_j by $J_{i\xi}$ and $J_{i\xi+1}$, respectively, in Theorem 2.3, it is clear that $J_{i\xi}$ precedes $J_{i\xi+1}$ in an optimal group schedule.

(Q. E. D.)

Corollary 2.4 For all jobs in a group, if the SPT schedule is coincident with the earliest-due-date (EDD) schedule, the optimal job sequence of this group is always the SPT schedule regardless of the group start time.

Considering that the optimal job sequences of the groups for which Theorem 2.6 does not hold are dependent on the group start time of each group, Algorithm II for determining the near optimal group sequence and job sequences that minimize the total tardiness is developed as follows.

< Algorithm II for determining a near optimal group schedule >

Step 1. Make the SPT schedules for job sequences for all groups. Go to Step 2.

Step 2. Determine a group sequence by using Algorithm I. Go to Step 3.

Step 3. Rearrange job sequences for the groups for which Theorem 2.6 does not hold; that is, compute the group start times in the group sequence derived and the revised due dates, then determine a job sequence in each group using an algorithm for a single job scheduling, such as Emmons' algorithm. Go to

Step 4.

Step 4. Calculate the total tardiness of this group schedule. Go to Step 5.

Step 5. If this schedule is the first one, then let this schedule be a temporary optimal group schedule, and go back to Step 2. Otherwise, go to Step 6.

Step 6. If this schedule is better than the temporary optimal group schedule, let this schedule be the new temporary optimal group schedule and go back to Step 2. Otherwise, terminate. The temporary optimal group schedule is a near optimal one.

2.3.4 Numerical examples

For production data shown in Table 2.1, the optimal group schedules can be determined by using Algorithms I and II. We shall demonstrate each case.

Table 2.1 Production data

(units: min)

Group	G_i	G_1			G_2	G_3		G_4		G_5			G_6	
Job	$J_{i\xi}$	J_{11}	J_{12}	J_{13}	J_{21}	J_{31}	J_{32}	J_{41}	J_{42}	J_{51}	J_{52}	J_{53}	J_{61}	J_{62}
Job processing time	$p_{i\xi}$	1	2	4	9	5	6	7	8	3	5	8	8	9
Due date	$d_{i\xi}$	46	20	38	85	40	32	50	57	70	26	27	39	51
Group setup time	S_i	1			1	2		1		1			2	
Group processing time	Q_i	8			10	13		16		12			12	

Example 1. (In the case of predetermined job sequence)

Step 1. G_2 is last, since d_{21} (=85) > Q (=83). Remove G_2.

Step 2. No result.

Step 3. G_4 is last, since G_1 and G_3 precede G_4, and G_5 and G_6 precede G_4. For example, it is shown as follows that G_1 precedes G_4. First, compute $d_{i\xi} + \sum_{\nu=\xi+1}^{n_i} p_{i\nu}$ for G_1 and G_4, then

$$\begin{cases} d_{11} + p_{12} + p_{13} = 52 \\ d_{12} + p_{13} = 24 \\ d_{13} = 38 \end{cases} \qquad \begin{cases} d_{41} + p_{42} = 58 \\ d_{42} = 57 \end{cases}$$

Since n_1 (=3) > n_4 (=2), by Theorem 2.3 two inequalities ($d_{12} + p_{13} > d_{42}$, $d_{13} > p_{41} + d_{42}$) imply that G_1 precedes G_4. Remove G_4, then we have now reduced the problem to a four-group problem $\{G_1, G_3, G_5, G_6\}$.

Step 4. $G \neq \phi$, go to Step 5.

Step 5. $D = \{G_1, G_5\}$, since it is shown by using Theorem 2.3 that G_1 precedes G_3 and G_6.

Step 6. D contains two groups, so go to Step 7.

Step 7. No result.

Step 8. Branch on the assumptions that (i) G_1 precedes G_5 and that (ii) G_5 precedes G_1.

Step 9. Select case (i) (node 1), then go back to Step 5.

Step 5. $D = \{G_1\}$.

Step 6. G_1 is first. Remove G_1, set (d_{31}, d_{32}, d_{51}, d_{52}, d_{53}, d_{61}, d_{62}) to (32, 24, 62, 18, 19, 31, 43) and go back to Step 4.

Step 4. $G = \{G_3, G_5, G_6\}$.

Step 5. $D = \{G_3, G_5\}$ since G_3 precedes G_6.

Step 6. D contains two groups, so go to Step 7.

Step 7. G_5 is first, since it is shown by using Theorem 2.4 that G_5 precedes G_3. Remove G_5, set (d_{31}, d_{32}, d_{61}, d_{62}) to (15, 9, 14, 28), and go back to Step 4.

Step 4. $G = \{G_3, G_6\}$.

Step 5. $D = \{G_3\}$ since G_3 precedes G_6.

The optimal group sequence (given that G_1 precedes G_5) is G_1-G_5-G_3-G_6-G_4-G_2, with T = 52 min. Go to Step 9 and select case (ii) (node 2). Then we have another optimal group sequence (given that G_5 precedes G_1), G_5-G_1-G_3

$-G_6-G_4-G_2$, with T = 53 min. Thus, evaluating only two schedules, the optimal group sequence is determined as $G_1-G_5-G_3-G_6-G_4-G_2$, with T = 52 min.

For large problems, determination of optimal group sequences may require a good deal of branching and evaluation of the schedules. The success of the algorithm will depend on the power of the theorems, and the degree to which branching is minimized. Whether the theorems hold or not depends on the structure of the data —— job processing times, due dates, and the number of jobs in each group. If we set the due dates sooner, then the optimal group sequence can be obtained easily because Theorem 2.3 is more useful. When the due dates of jobs in groups tend to be longer relative to the total processing time Q, we anticipate greater difficulty with the problems. In such problems, however, Theorems 2.4 and 2.5 tend to be more useful.

Example 2. (In the case that job sequences are not predetermined)

Step 1. The job sequence in each group in Table 2.1 is already the SPT schedule.

Step 2. The optimal group sequence is $G_1-G_5-G_3-G_6-G_4-G_2$ as shown in example 1.

Step 3. The groups in which job sequences depend on their group start times are G_1, G_3, and G_5. The group start times of these groups are $t_1 = 0$, $t_3 = 25$, and $t_5 = 8$.

(i) The job sequence of G_1 is arbitrary, with $T_1 = 0$. Suppose $J_{11}-J_{12}-J_{13}$.

(ii) The job sequence of G_3 is $J_{32}-J_{31}$, with $T_3 = 1$.

(iii) The job sequence of G_5 is arbitrary, with $T_5 = 0$. Suppose $J_{51}-J_{52}-J_{53}$.

Step 4. This group schedule is given as shown in Table 2.2 and T = 47 min.

Step 5. Go back to Step 2 and determine a new group schedule in the same way.

Since this new schedule is coincident with the schedule obtained before, then the near optimal group schedule is shown in Table 2.2.

Table 2.2 A near optimal group schedule

Group sequence	G_1*			G_5*			G_3		G_6		G_4		G_2	Total tardiness
Job sequence	J_{11}	J_{12}	J_{13}	J_{51}	J_{52}	J_{53}	J_{32}	J_{31}	J_{61}	J_{62}	J_{41}	J_{42}	J_{21}	47 min

* Job sequences in these groups are arbitrary.

To test the effectiveness of this algorithm, ten examples were investigated, each consisting of six to eight groups with one to three jobs in each group. For these problems, all the group schedules determined by using this algorithm were optimal and the number of repetitions of Step 2 through Step 5 was one in all cases.

2.4 Conclusions

(1) The single-stage group scheduling model which is fundamental in group production scheduling was constructed.

(2) The group scheduling model was analyzed under two kinds of criteria — the minimum mean flow time and the minimum weighted mean flow time, and two theorems which give the optimal group schedules (optimal sequences of groups and jobs) were proved.

(3) Under the criterion of minimizing the total tardiness, several theorems were given for determining the relative order of pairs of groups. With the use of them, the algorithms for determining the optimal and the near optimal group schedules were proposed and numerical examples were shown.

CHAPTER 3 GROUP SCHEDULING WITH SEQUENCE-DEPENDENT SETUP TIMES ON A SINGLE PRODUCTION STAGE

3.1 Introduction

When a job (part) is processed on a machine, time is required to setup the machine for the next job before the actual machining of the job. Usually, setup times are assumed to be independent of the sequence of jobs and are included in the processing times. In many realistic problems, however, setup times depend on the type of job just completed as well as on the job to be processed on a machine.

In the conventional scheduling problem where only one group is involved, the problem with sequence-dependent setup times is a formidable one even for the simple criterion of minimizing the total elapsed time. It is well known that the problem corresponds to the so-called "traveling salesman problem." Although no simple algorithm for solving the traveling salesman problem is known, several solution procedures that will obtain optimal solutions to problems of modest size and approximate solutions to larger problems have recently been developed. The optimizing procedures involve dynamic programming[1,2] and the branch-and-bound method,[3] both being general purpose methodologies. A heuristic procedure[4,5] is one of the methods which give near optimal solutions.

A problem involving more complex measures, such as minimizing the mean flow time, becomes a more formidable one. This problem can be shown to correspond to a quadratic assignment problem which can only be solved for problems of small size by resorting to the general purpose methodologies such as dynamic programming and the branch-and-bound method.

The provious chapter deals with the group scheduling problems with sequence-independent setup times. In this chapter, an attempt is made to solve group scheduling problems in which group setup times are dependent on the sequence of groups.[6,7] In the model, job setup times are assumed to be sequence-independent and are included in job processing times.

In the next section, the problem for the criterion of minimization of the total elapsed time is also shown to be reduced to the traveling salesman problem. For the problem of minimizing the mean flow time, two kinds of solution approaches are offered in Section 3.3. First, with the use of dynamic programming, the optimal group schedule is determined. Second, a simple branch-and-bound algorithm is developed for solving the problem. In the last section, an analysis of the optimal group scheduling for the minimum total tardiness is made and an efficient algorithm using a branch-and-bound method is proposed to find the optimal group schedule. The effectiveness of the algorithm proposed is verified by numerical examples.

3.2 Minimizing total elapsed time

In the basic single-stage problem, the total elapsed time, the time to complete all jobs within all groups, is a constant as shown in equation (2.1). With sequence-dependent setup times, the total elapsed time, however, depends on which sequences of groups and jobs are chosen.

Let S_{ij} $(i, j = 1, 2, \ldots, N)$ denote the group setup time required for group G_j after G_i is completed and let $p_{i\xi}$ and $d_{i\xi}$ $(i = 1, 2, \ldots, N, \xi = 1, 2, \ldots, n_i)$ denote the job processing time including the job setup time and the due date of job $J_{i\xi}$ $(i = 1, 2, \ldots, N, \xi = 1, 2, \ldots, n_i)$, as before.

The total elapsed time is given by

$$F_{max} = \sum_{i=1}^{N} S_{(i-1)(i)} + \sum_{i=1}^{N} \sum_{\xi=1}^{n_i} p_{(i)(\xi)} \tag{3.1}$$

where $S_{(0)(1)}$ is the setup time required to bring the machine from idleness to a state ready to process the first group in sequence.

Since the second summation is a constant, the problem of minimizing the total elapsed time is equivalent to minimizing the first summation.

This sum indicates the total group setup time in the full sequence of groups, beginning and ending in the idle state.

The problem of minimizing the sum of the group setup times corresponds to the so-called "traveling salesman problem." In the original formulation, a salesman must visit each of n cities once and only once and return to his point of origin, and do so such that the total travel distance is minimized. In the scheduling problem, a job corresponds to each city, and the group setup time S_{ij} corresponds to the distance between cities i and j. By defining a hypothetical group G_0 such that

$$P_0 = 0, \quad S_{0i} = S_i, \quad \text{and } S_{i0} = 0 \tag{3.2}$$

where S_i is the group setup time when G_i is processed first in sequence, and letting G_0 be the starting point, the scheduling problem stated above actually becomes a $(n+1)$-city problem. In the original traveling salesman problem, of course, the distance matrix $[S_{ij}]$ is symmetric, that is, $S_{ij} = S_{ji}$. This is not always the case in the scheduling problem. However, the nonsymmetricity of the matrix does not appear to make the problem significantly more difficult to solve.

From the above analysis, the single-stage group scheduling with sequence-dependent setup times and of minimizing the total elapsed time can be solved by several solution procedures which have been developed for the traveling salesman problem.

3.3 Minimizing mean flow time

In the case of sequence-dependent group setup times, the problem of minimizing the mean flow time is a more challenging one to solve. The mean flow time is obtained in much the same way as equation (2.4), as follows:

$$\bar{F} = \frac{1}{M}\{\sum_{i=1}^{N} n_{(i)} \sum_{j=1}^{i-1} (S_{(j-1)(j)} + P_{(j)}) + \sum_{i=1}^{N} n_{(i)} S_{(i-1)(i)} + \sum_{i=1}^{N} \sum_{\xi=1}^{n_i} \sum_{\nu=1}^{\xi} p_{(i)(\nu)}\} \tag{3.3}$$

where $P_{(j)} = \sum_{\xi=1}^{n_j} p_{(j)(\xi)}$ and $M = \sum_{i=1}^{N} n_i$.

Minimizing the above equation corresponds to minimizing the following one, which represents the total flow time.

$$F = \sum_{i=1}^{N} n_{(i)} \sum_{j=1}^{i-1} (S_{(j-1)(j)} + P_{(j)}) + \sum_{i=1}^{N} n_{(i)} S_{(i-1)(i)} + \sum_{i=1}^{N} \sum_{\xi=1}^{n_i} \sum_{\nu=1}^{\xi} p_{(i)(\nu)} \tag{3.4}$$

The first and second summations of the above equation are concerned with the group sequence, and the third one is concerned with the job sequence. Since $P_{(j)}$, the total processing time of all jobs in each group is a constant, independent of the job sequence, the group sequence and the job sequences can be handled independently in minimizing equation (3.4) The third summation is minimized by ordering the jobs in each group in order of nondecreasing job processing time. Thus, the optimal job sequence is determined to be SPT sequencing for each group. The problem of minimizing the sum of the first and second terms is a formidable one. For determination of the optimal group sequence, two general purpose methodologies, such as dynamic programming and the branch-and-bound method, are employed.

3.3.1 Dynamic programming approach

Dynamic programming is a technique for solving a special class of optimization problems called multistage decision processes. It has evolved primarily on the basis of R. Bellman's works which date back to the early 1950's.[8] Unlike other optimization methods, such as linear programming, a specific mathematical form for the class of optimization problems which dynamic programming can solve cannot be presented. The possibility of applying the dynamic programming method depends on a successful formulation of the problem in terms of a multistage decision process.

Applications of the dynamic programming method to the scheduling problem have been made by Bellman,[1] and by Held and Karp.[2] This procedure is perhaps more general than the branch-and-bound method which will be used to solve the current problem in later sections.

With some slight modifications, the dynamic programming approach can be adapted to determine an optimal group sequence for the problem.[7] Suppose G_0 is a hypothetical group having the properties of equation (3.2) and is the first group in sequence. Let G_i denote one of the groups, not equal to G_0, and K denote a set consisting of k groups, not G_i and G_0. Furthermore, let G denote the set of all groups, excluding G_0. Now define the following function:

$f(G_i, K)$ = the minimum total flow time from the beginning of group G_i to the end of G_0, with intermediate processing of k groups in K

Then, the total processing time of the optimal group sequence is given by $f(G_0, G)$. Introduce the following notations:

$$\left.\begin{aligned} Q_{ij} &= S_{ij} + P_j, \quad R_k = \sum_{k\in K} n_k - n_j \\ E_{ij} &= n_i S_{ij} + \min \sum_{\xi=1}^{n_j} \sum_{\nu=1}^{\xi} p_{j\nu} \end{aligned}\right\} \qquad (3.5)$$

where $\min \sum_{\xi=1}^{n_j} \sum_{\nu=1}^{\xi} p_{j\nu}$ is achieved when the jobs in each group are ordered by SPT sequencing.

A dynamic programming formulation is made by the principle of optimality, as follows: (See Fig. 3.1.)

$$\left.\begin{aligned} f(G_i, K) &= \min_{j\in K} \{E_{ij} + Q_{ij} R_k + f(G_j, K - \{j\})\} \\ f(G_i, \phi) &= S_{i0} = 0 \end{aligned}\right\} \qquad (3.6)$$

By using this recursive relation, the optimal group sequence is determined by first considering sets K of size 1, then sets K of size 2, and so on until $f(G_0, G)$ at the final stage is obtained.

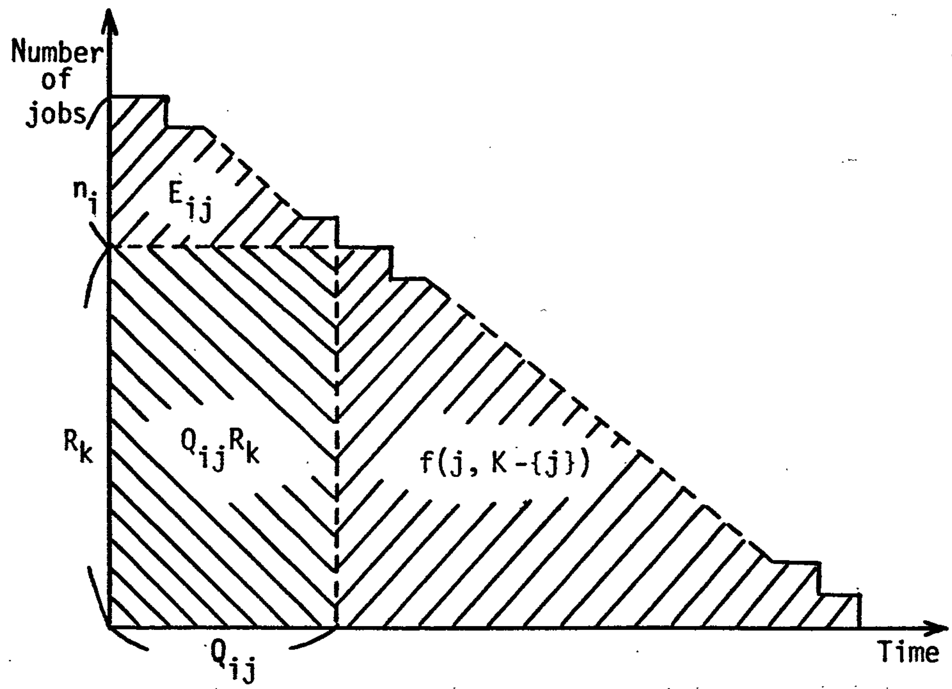


Fig. 3.1 The total flow time from the beginning of G_i to the end of G_0, with intermediate processing of k groups in K

In general, at each stage, there are N ways of selecting group G_i, and for each of these, there are $\binom{N-1}{k}$ ways of selecting the groups of K. Therefore, the total number of computations of equation (3.6) required to determine an optimal group sequence is given by

$$\sum_{k=1}^{N-1} N\binom{N-1}{k} + N \tag{3.7}$$

3.3.2 Branch-and-bound approach

The branch-and-bound method is one useful method for solving many combinatorial problems,[9] and is particularly suited to well-structured problems with integer constraints on the variables. Like dynamic programming it does not deal with a specific mathematical framework nor does it follow the conventional iterative idea of an optimization process. Its aim is to conduct a reduced search over all possible solutions, the reduction being dependent on how well the problem structure can be exploited.

As its name implies, the branch-and-bound method consists of two fundamental procedures —— branching and bounding procedures. Branching is the process of partitioning a large problem into two or more subproblems by a specified rule, and bounding is the process of calculating a lower bound (in the case of minimization) for the solution to each subproblem generated in the branching procedure. After each partitioning, those subproblems with bounds that exceed the performance measure of a known feasible solution are excluded from further partitioning. The partitioning is repeated until a feasible solution is found such that its performance measure is no greater than the bound for any subproblem.

With some success, the branch-and-bound method has been employed to solve several scheduling problems.[10] The application of this method to the traveling salesman problem was successfully made by Little et al.[3]

Now apply this method to the problem of determining the optimal group sequence. The total flow time represented by equation (3.4) can be transformed into the following one:

$$F = \sum_{i=1}^{N} n_{(i)} \sum_{j=1}^{i} Q_{(j-1)(j)} - \sum_{i=1}^{N} n_{(i)} P_{(i)} + \sum_{i=1}^{N} \sum_{\xi=1}^{n_i} \sum_{\nu=1}^{\xi} p_{(i)(\nu)} \tag{3.8}$$

where $Q_{(j-1)(j)} (= S_{(j-1)(j)} + P_{(j)})$ is the group processing time of $G_{(j)}$ in the case that $G_{(j)}$ is processed after the completion of $G_{(j-1)}$.

The first summation is dependent on the sequence of groups, while the second and third ones are independent of which group sequence is decided. That is, the second and third summations are not concerned with the group sequence. Hence, the problem of finding the optimal group sequence corresponds to determining the sequence of groups so as to minimize the first summation of equation (3.8). In a sense, this sum represents total group flow time weighted by the number of jobs in each group.

The branching and bounding procedures for this problem are as follows:

Branching procedure: The set of permutations of group indices is partitioned into several subsets. By this procedure, nodes, each of which represents a subset of the group sequence, are repeatedly created. Let N_r be a node at which the sequence of r groups is determined: $N_r = \{G_{(1)}, G_{(2)}, \ldots, G_{(r)}\}$. Branching from this node consists of taking each of $(N-r)$ unallocated groups and placing it next in the sequence determined. Then, new $(N-r)$ nodes, N_{r+1}, which have the sequence, $G_{(1)}-G_{(2)}- \cdots -G_{(r+1)}$, are created.

Bounding procedure: The lower bound on the first summation of equation (3.8) at N_r is estimated by

$$F(N_r) = F_1(N_r) + F_2(N_r) \tag{3.9}$$

where $F_1(N_r)$ and $F_2(N_r)$ are the total weighted flow times for the groups sequenced and for the groups not yet sequenced, respectively.

Clearly, $F_1(N_r)$ is given by

$$F_1(N_r) = \sum_{i=1}^{r} n_{(i)} \sum_{j=1}^{i} Q_{(j-1)(j)} \tag{3.10}$$

In order to calculate $F_2(N_r)$, construct a square matrix of order $(N+1)$ which is defined as

$$[Q_{ij}] = \begin{bmatrix} 0, & Q_{01}, & Q_{02}, & \ldots, & Q_{0N} \\ 0, & \infty, & Q_{12}, & \ldots, & Q_{1N} \\ 0, & Q_{21}, & \infty, & \ldots, & Q_{2N} \\ \vdots & \vdots & \vdots & & \vdots \\ 0, & Q_{N1}, & Q_{N2}, & \ldots, & \infty \end{bmatrix} \tag{3.11}$$

From this matrix, select the $(N-r)$ smallest values from among the elements in each of the $(N-r)$ columns, excluding ones in the first row and the rows corresponding to the first $(r-1)$ groups already sequenced. Compute the ratios of the group processing time selected to the number of the jobs in the group, then order them in nondecreasing order. Denoting the group processing time selected and the number of jobs in the group by $Q'_{(i)}$ and $n'_{(i)}$ $(i = r+1, r+2, \ldots, N)$, respectively, we have the following order:

$$\frac{Q'_{(r+1)}}{n'_{(r+1)}} \leq \frac{Q'_{(r+2)}}{n'_{(r+2)}} \leq \cdots \leq \frac{Q'_{(N)}}{n'_{(N)}}$$

Thus, $F_2(N_r)$ is given by

$$F_2(N_r) = \sum_{i=r+1}^{N} n'_{(i)} \left(t_{(r+1)} + \sum_{j=r+1}^{i} Q'_{(j)}\right) \tag{3.12}$$

where $t_{(r+1)} = \sum_{i=1}^{r} Q_{(i-1)(i)}$.

The total weighted group flow time is a nondecreasing function of the completion time of each group, and is minimized by ordering the groups in nondecreasing order of the ratios of the group processing time to the number of jobs in the group. Therefore, equation (3.12) gives a lower bound on the total weighted flow time for $(N-r)$ groups not yet sequenced.

The previous analysis leads to the following branch-and-bound algorithm for determining an optimal group schedule.

< Optimizing algorithm for the minimum mean flow time >

Step 1. Order the jobs in each group by SPT sequencing. Go to Step 2.

Step 2. Let the level of the node $r = 0$ and the least feasible total flow time $F^* = \infty$. Go to Step 3.

Step 3. Branch the node into $(N-r)$ nodes by placing each of the not yet allocated groups next in the sequence determined. Set $r = r+1$, and go to Step 4.

Step 4. Calculate the lower bound $F(N_r)$ for each of the new nodes. Go to Step 5.

Step 5. Select the node having the minimum lower bound from among those newly created in Step 3 when $F^* = \infty$, or from among all nodes being active when $F^* \neq \infty$.(In the case of a tie, choose the node with the largest value of r. Break the tie arbitrarily for the same r.) Let the level of the node selected be r and $F^*(N_r) = F(N_r)$. Go to Step 6.

Step 6. If $r < N$, then go back to Step 3. Otherwise, go to Step 7.

Step 7. If $F^*(N_r) < F^*$, then $F^* = F^*(N_r)$, so go back to Step 5. Otherwise, the group sequence associated with the node having F^* is optimal. Stop.

As a simple illustration, consider the 6-group, 15-job group scheduling problem of minimizing the mean flow time, as shown in Tables 3.1 and 3.2.

Table 3.1 Group setup times (units: min)

Group No.	0	1	2	3	4	5	6
0	-	7	9	11	8	6	10
1	0	-	8	10	13	11	9
2	0	7	-	9	12	8	7
3	0	11	14	-	10	9	8
4	0	13	11	10	-	7	9
5	0	12	8	9	10	-	11
6	0	14	9	9	8	7	-

Table 3.2 Job processing times and due dates

(units: min)

Group	G_1			G_2		G_3		G_4		G_5			G_6		
Job	J_{11}	J_{12}	J_{13}	J_{21}	J_{22}	J_{31}	J_{32}	J_{41}	J_{42}	J_{51}	J_{52}	J_{53}	J_{61}	J_{62}	J_{63}
Job processing time	4	5	7	5	10	11	13	8	10	6	10	15	7	9	12
Due date	25	72	32	65	110	122	72	57	93	140	98	52	47	88	52

In Table 3.2, the job sequences are already ordered by SPT sequencing. Hence, an optimal decision as to the group sequence is to be made. The square matrix $[Q_{ij}]$ of the group processing times including the setup times, is given in Table 3.3. By using the optimizing algorithm proposed, an optimal group schedule is determined as $G_1(J_{11}-J_{12}-J_{13})-G_2(J_{21}-J_{22})-G_6(J_{61}-$

Table 3.3 Matrix $[Q_{ij}]$

Group No.	0	1	2	3	4	5	6
0	-	23	24	35	26	37	38
1	0	-	23	34	31	42	37
2	0	23	-	33	30	39	35
3	0	27	29	-	28	40	36
4	0	29	27	34	-	38	37
5	0	28	23	33	28	-	39
6	0	30	24	33	26	38	-

$J_{62}-J_{63})-G_4(J_{41}-J_{42})-G_5(J_{51}-J_{52}-J_{53})-G_3(J_{31}-J_{32})$ with the mean flow time of 85.6* min. The branching tree of this problem is shown in Fig. 3.2.

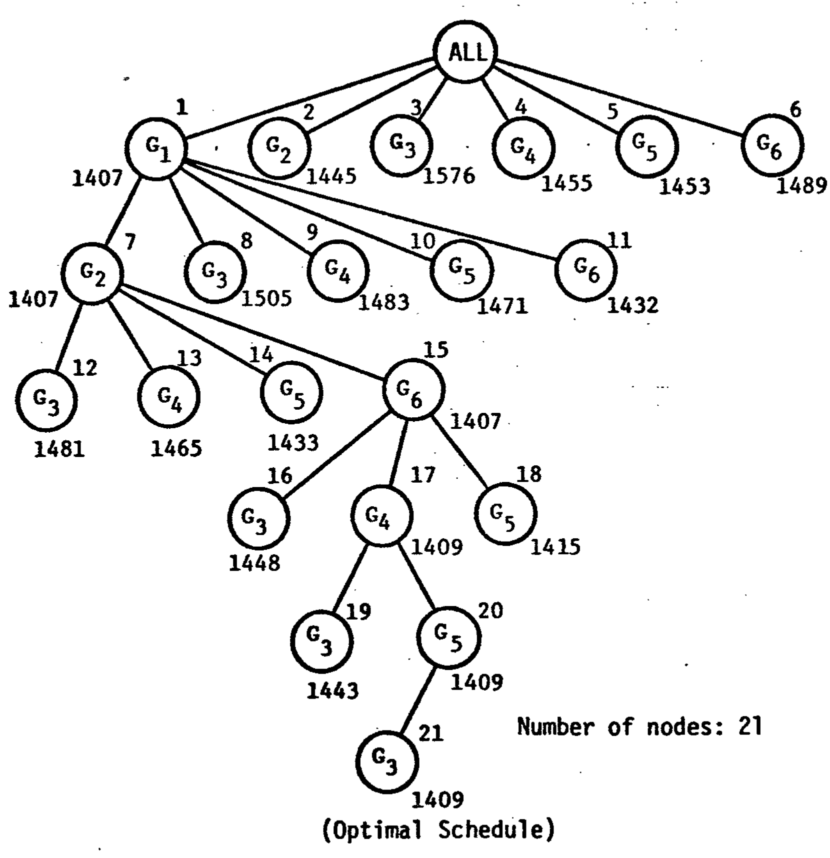


Fig. 3.2 The branching tree for the minimum-mean-flow-time problem

* This is calculated as follows:

$$\sum_{i=1}^{N} n_{(i)}P_{(i)} = 339, \sum_{i=1}^{N}\sum_{\eta=1}^{n_i}\sum_{\nu=1}^{\eta} p_{(i)(\nu)} = 214. \text{ Hence, } \bar{F} = \frac{1409-339+214}{15} = 85.6.$$

The figures just below the nodes indicate the lower bounds on the total weighted group flow time. The CPU time required to determine the optimal group schedule is within 1 second with the TOSBAC model 140 computer.

3.4 Minimizing total tardiness

In the group scheduling problem of minimizing the total tardiness, two decisions as to the sequences of groups and jobs in each group cannot be made independently of each other unlike the case of minimizing the mean flow time. For determining an optimal group schedule, the branch-and-bound method is employed in the same way as in the previous section. In order to increase the efficiency of the branch-and-bound algorithm, two theorems which specify the relative order of pairs of jobs within the same group in an optimal schedule are offered. Then, based on the branch-and-bound method, the optimizing algorithm which incorporates the two theorems is developed for the optimal group scheduling, and a numerical example is shown.

The total tardiness of all jobs in all groups is expressed by the following equation, similar to equation (2.8) in Chapter 2.

$$T = \sum_{i=1}^{N} T_{(i)} = \sum_{i=1}^{N} \sum_{\xi=1}^{n_i} \max\{\sum_{j=1}^{i-1} (S_{(j-1)(j)} + P_{(j)} + S_{(i-1)(i)} + \sum_{\nu=1}^{\xi} p_{(i)(\nu)} - d_{(i)(\xi)},\ 0\} \tag{3.13}$$

3.4.1 Theorems for job sequence

The following theorems that establish the relative order in which pairs of jobs are processed in an optimal group schedule are of use for the reduction of the number of subproblems generated in the process of branching procedure.

Theorem 3.1 For any two jobs $J_{i\xi}$ and $J_{i\eta}$ with $p_{i\xi} \leq p_{i\eta}$, if $d_{i\xi} \leq d_{i\eta}$, then $J_{i\xi}$ precedes $J_{i\eta}$ in an optimal group schedule, irrespective of the schedule time at which scheduling is done for groups and jobs not yet sequenced.

Proof. If the schedule time is set at t, then the revised due dates of $J_{i\xi}$ and $J_{i\eta}$ are $d_{i\xi} - t$ and $d_{i\eta} - t$, respectively. Since $d_{i\xi} \le d_{i\eta}$, then $d_{i\xi} - t \le d_{i\eta} - t$. Supposing $n_i = n_j = 1$, and replacing G_i and G_j by $J_{i\xi}$ and $J_{i\eta}$, respectively, in Theorem 2.3, it follows that $J_{i\xi}$ precedes $J_{i\eta}$.

(Q. E. D.)

Theorem 3.2 For any two jobs $J_{i\xi}$ and $J_{i\eta}$ with $p_{i\xi} \le p_{i\eta}$, if the schedule time $t \ge d_{i\xi} - p_{i\eta}$, then $J_{i\xi}$ precedes $J_{i\eta}$ in an optimal group schedule.

Proof. Let S be any schedule in which for two jobs in G_i, $J_{i\eta}$ precedes $J_{i\xi}$. Consider a schedule S' that differs from S only in that $J_{i\xi}$ and $J_{i\eta}$ are interchanged. All jobs in G_i between $J_{i\eta}$ and $J_{i\xi}$ are advanced in time by the amount of $(p_{i\eta} - p_{i\xi}) \ge 0$, which does not in the least increase the total tardiness. Denote by X and Y the times at which $J_{i\eta}$ begins and $J_{i\xi}$ ends, respectively, in S (see Fig. 3.3). Then, from the condition, $X \ge t$. It can be

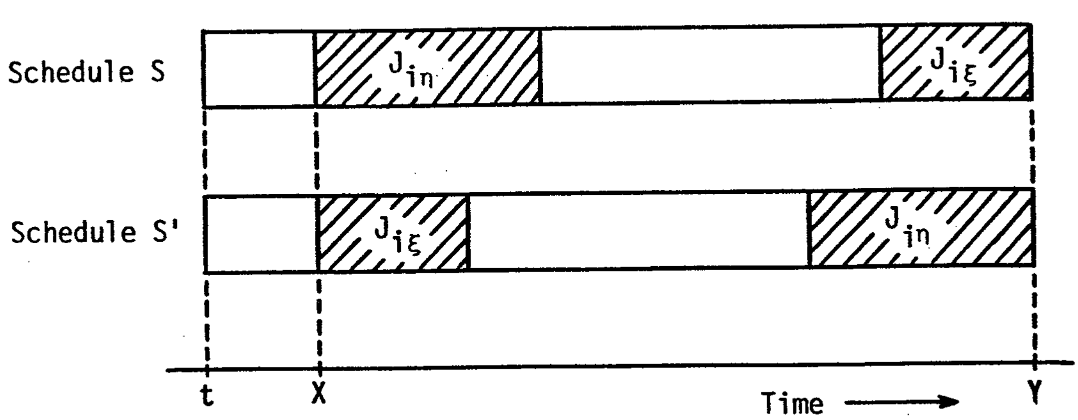


Fig. 3.3 The effect of interchanging two jobs

shown that interchanging the two jobs must decrease, or possibly leave unchanged, the total tardiness. The decrease of the tardiness of $J_{i\xi}$ is $\Delta T^-_{i\xi} = Y - \max(X + p_{i\xi}, d_{i\xi})$, since $d_{i\xi} \le t + p_{i\eta} \le X + p_{i\eta} \le Y$.

The increase of tardiness of $J_{i\eta}$ is given by

$$\Delta T_{i\eta}^{+} = \begin{cases} 0, & \text{if } d_{i\eta} \geq Y \\ Y - \max(X + p_{i\eta}, d_{i\eta}), & \text{otherwise} \end{cases}$$

Hence, if $d_{i\eta} \geq Y$, clearly $\Delta T_{i\xi}^{-} \geq \Delta T_{i\eta}^{+}$. If $d_{i\eta} < Y$, then $\Delta T_{i\xi}^{-} - \Delta T_{i\eta}^{+} =$ $\max(X + p_{i\eta}, d_{i\eta}) - \max(X + p_{i\xi}, d_{i\xi})$. This gives $\Delta T_{i\xi}^{-} \geq \Delta T_{i\eta}^{+}$, since $X + p_{i\eta} \geq d_{i\xi}$ and $X + p_{i\eta} \geq X + p_{i\xi}$.

(Q. E. D.)

3.4.2 Application of branch-and-bound method to group scheduling

Since, in group scheduling, both optimal group and job sequences must be determined simultaneously, a branch-and-bound procedure of a new type is required. The first application of the method was made to the group scheduling with sequence-independent setup times by Nakamura and Hitomi.[11) They offered the basic idea of branching the scheduling problem into sub-problems. In this subsection, an improved branching procedure using the two theorems proved before is developed and an efficient formula for the lower bound on the total tardiness is offered.

The basic branching procedure for the group scheduling is as follows: In group scheduling, branching of groups and branching of jobs are both required since optimal decisions are made as to the sequences of groups and jobs in each group. Eventually, there occur two kinds of nodes ——"group node" and "job node." Basically, the branching of groups is made firstly by taking each of the unsequenced groups in turn, and placing it next in the permutation of groups determined. Then, in the same way, jobs are branched from each of the group nodes created. The procedure of branching jobs in the current group is repeated until the positions of all jobs in the group are determined. After that, new group nodes are created by branching unallocated groups at each of the nodes. The process of branching groups and jobs is shown in Fig. 3.4.

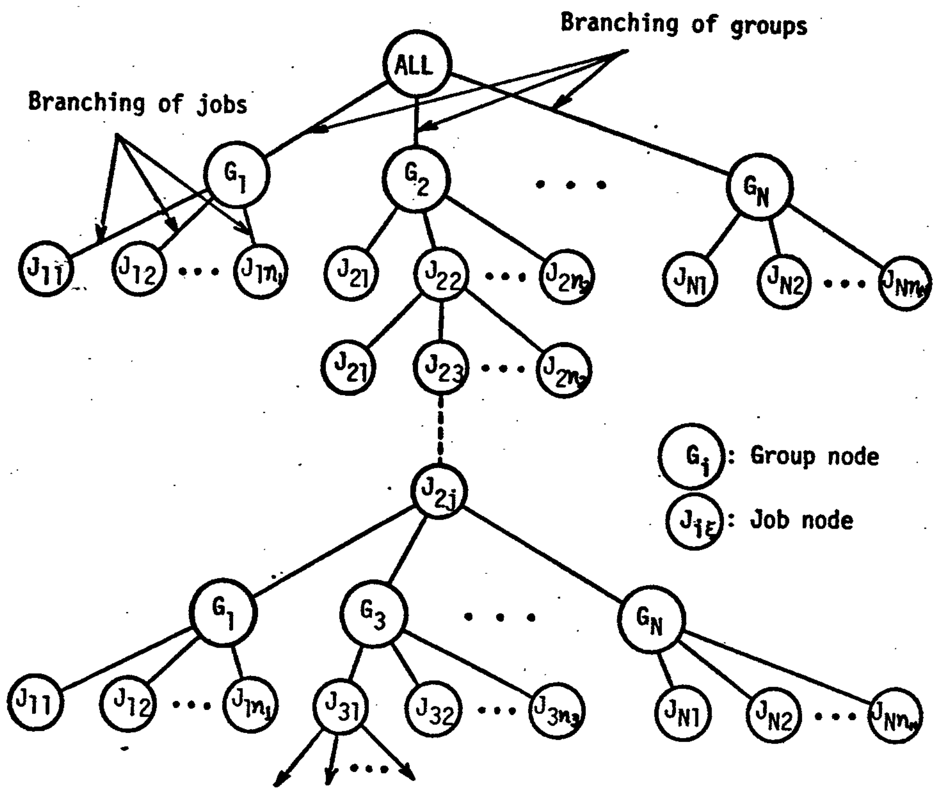


Fig. 3.4 The branching process for group scheduling problem

The bounding procedure is a process of calculating the lower bound on the solution of the subproblem represented by each job node. The formula for the lower bound depends on the scheduling criterion employed.

3.4.3 Optimizing algorithm based on branch-and-bound method

The branching and bounding procedures for determining the optimal group schedule minimizing the total tardiness are as follows:

(1) Branching procedure

Basically, the branching procedure for the problem of minimizing the total tardiness is similar to the basic one for group scheduling, but differs from it in that branching of jobs is made according to each job's precedence relations which are specified by Theorems 3.1 and 3.2. Let N_r be a group node at which the sequence of r groups is specified: $N_r = \{G_{(1)}, G_{(2)}, \ldots, G_{(r)}\}$ and N_{rs} be a job node at which s jobs in group $G_{(r)}$ are allocated: $N_{rs} = \{J_{(r)(1)}, J_{(r)(2)}, \ldots, J_{(r)(s)}\}$. Then N_r and N_{rs} are called r group-level node and s-level node, respectively. The branching of jobs at N_{rs} is made by the following procedure.

< The branching procedure for jobs >

(i) Select a job, $J_{(r)\alpha}$, from among jobs not yet sequenced in $G_{(r)}$ at N_{rs}, and go to Step (ii). If all jobs in $G_{(r)}$ are allocated, then stop.

(ii) With the use of Theorem 3.1 select the jobs which precede $J_{(r)\alpha}$. When none of the jobs precedes $J_{(r)\alpha}$, go to Step (iii). If all jobs which precede $J_{(r)\alpha}$ are already sequenced, then go to Step (iii). Otherwise, go back to Step (i).

(iii) Search the jobs which precede $J_{(r)\alpha}$ by letting the start time of $J_{(r)\alpha}$ be the schedule time and applying Theorem 3.2 to $J_{(r)\alpha}$ and others. If no jobs preceding $J_{(r)\alpha}$ exist, then go to Step (iv). Otherwise, go to Step (iv), or go back to Step (i) according to whether or not all the jobs are already sequenced, respectively.

(iv) Create a job node $N_{rs+1} = \{J_{(r)(1)}, J_{(r)(2)}, \ldots, J_{(r)\alpha}\}$ of job level of $(s+1)$ by placing $J_{(r)\alpha}$ next in the permutation already sequenced and go back to Step (i).

By using this procedure, a large amount of reduction of branches generated can be expected; hence, the time needed to determine an optimal group schedule will be reduced.

(2) Bounding procedure

The computation of the lower bound on the total tardiness is performed on each of the job nodes generated by the branching procedure. The lower bound at N_{rs} is estimated by

$$T(N_{rs}) = T_1(N_{rs}) + T_2(N_{rs}) + T_3(N_{rs}) \tag{3.14}$$

where $T_1(N_{rs})$, $T_2(N_{rs})$, and $T_3(N_{rs})$ are the total tardinesses for groups and jobs already sequenced, for jobs not yet allocated in $G_{(r)}$, and for groups not yet assigned, respectively. They are calculated in the following way:

The first term of the above equation is obvious, as follows:

$$T_1(N_{rs}) = \sum_{i=1}^{r-1}\sum_{\xi=1}^{n_i}\max\{\sum_{j=1}^{i-1}(S_{(j-1)(j)} + P_{(j)}) + S_{(i-1)(i)} + \sum_{\nu=1}^{\xi}P_{(i)(\nu)} - d_{(i)(\xi)},\ 0\} + \sum_{\xi=1}^{s}\max\{\sum_{j=1}^{r-1}(S_{(j-1)(j)} + P_{(j)}) + S_{(r-1)(r)} + \sum_{\nu=1}^{\xi}P_{(r)(\nu)} - d_{(r)(\xi)},\ 0\} \tag{3.15}$$

The second term for $(n_{(r)} - s)$ jobs not yet sequenced in $G_{(r)}$ is given by

$$T_2(N_{rs}) = \sum_{\xi=s+1}^{n_r}\max(C_{(r)(s)} + \sum_{\nu=s+1}^{\xi}p'_{(r)(\nu)} - d'_{(r)(\xi)},\ 0) \tag{3.16}$$

where $C_{(r)(s)}$ is the completion time of $J_{(r)(s)}$, and $p'_{(r)(\nu)}$ and $d'_{(r)(\nu)}$ $(\nu = s+1, s+2, \ldots, n_r)$ are, respectively, the processing times and the due dates of jobs not yet sequenced in $G_{(r)}$, which are ordered independently of each other, such that

$$p'_{(r)(s+1)} \le p'_{(r)(s+2)} \le \cdots \le p'_{(r)(n_r)}$$

$$d'_{(r)(s+1)} \le d'_{(r)(s+2)} \le \cdots \le d'_{(r)(n_r)}$$

The third term of equation (3.14) is calculated by the following steps:

(i) Construct a square matrix $[S_{ij}]$ of group setup times, which is similar to the matrix $[Q_{ij}]$ in the previous subsection. For this matrix, let $S'_{(r)(r+1)}$ be the smallest value among the elements in rth row excluding ones in the columns corresponding to the groups already sequenced. Furthermore, select the $(N-r)$ smallest values from among the elements in each of the $(N-r)$ columns excluding ones in the first row and the rows corresponding to groups already sequenced, and then order them such that

$$S'_{(r+1)(r+2)} \le S'_{(r+2)(r+3)} \le \cdots \le S'_{(N-1)(N)} \le S'_{(N)(N+1)}$$

(ii) Order the numbers of jobs for $(N-r)$ groups not yet sequenced such that

$$n'_{(r+1)} \geq n'_{(r+2)} \geq \cdots \geq n'_{(N)}$$

(iii) Order the processing times and the due dates for jobs not yet sequenced in nondecreasing order, respectively, irrespective of the groups to which the jobs belong.

(iv) Make $(N-r)$ hypothetical groups by grouping the processing times and the due dates by $n'_{(i)}$ items $(i=r+1, r+2, \ldots, N)$, respectively, as shown in Table 3.4.

Table 3.4 Hypothetical groups

Group	$G'_{(r+1)}$	$G'_{(r+2)}$	...	$G'_{(N)}$
Number of jobs	$n'_{(r+1)}$	$n'_{(r+2)}$	...	$n'_{(N)}$
Processing time	$p'_{(r+1)(1)}, \ldots$	$p'_{(r+2)(1)}, \ldots$	...	$p'_{(N)(1)}, \ldots$
Due date	$d'_{(r+1)(1)}, \ldots$	$d'_{(r+1)(2)}, \ldots$	...	$d'_{(N)(1)}, \ldots$
Group setup time	$S'_{(r)(r+1)}$	$S'_{(r+1)(r+2)}$	...	$S'_{(N-1)(N)}$

Then $T_3(N_{rs})$ is given by

$$T_3(N_{rs}) = \sum_{i=r+1}^{N} \sum_{\xi=1}^{n'_i} \max\{C_{(r)(n_r)} + \sum_{j=r+1}^{i-1} (S'_{(j-1)(j)} + P'_{(j)}) + S'_{(i-1)(i)} + \sum_{\nu=1}^{\xi} p'_{(i)(\nu)} - d'_{(i)(\xi)},\ 0\} \tag{3.17}$$

where $P'_{(j)} = \sum_{\xi=1}^{n'_j} p'_{(j)(\xi)}$.

The value of equation (3.14), the sum of the job tardinesses calculated by equations (3.15), (3.16), and (3.17), is a lower bound, since the total tardiness is a nondecreasing function of the completion time of each job.

Based on the previous analysis, the following algorithm is developed to find an optimal group schedule.

< Optimizing algorithm for the minimum total tardiness >

Step 1. Let the group level $r = 0$ and the least feasible total tardiness $T^* = \infty$. Go to Step 2.

Step 2. Create $(N - r)$ new group nodes N_{r+1}. Set $r = r + 1$, and go to Step 3.

Step 3. Let the job level $s = 1$ and by using the branching procedure for jobs, create new job nodes N_{rs} from each of the group nodes made. Go to Step 4.

Step 4. Calculate the lower bound $T(N_{rs})$ for each of the new job nodes by equation (3.14). Go to Step 5.

Step 5. Find the job node having min $T(N_{rs})$ from among the job nodes derived in Step 3, or 8 in the case of $T^* = \infty$, or from among all job nodes being active in the case of $T^* \neq \infty$. (In the case of a tie, select the node with the largest value of, first, r, and then s.) Let the group level and job level of the node be r and s, respectively, and $T^*(N_{rs}) = T(N_{rs})$. Go to Step 6.

Step 6. If $T^*(N_{rs}) < T^*$, then go to Step 7. Otherwise, the group schedule associated with the node having T^* is optimal. (T^* is the minimum total tardiness.) Stop.

Step 7. If $s < n_{(r)}$, then go to Step 8. Otherwise, go to Step 9.

Step 8. Set $s = s + 1$, and by using the branching procedure for jobs, create new job nodes N_{rs} from the current job node. Go back to Step 4.

Step 9. If $r < N$, then go back to Step 2. Otherwise, $T^* = T^*(N_{rs})$, so go back to Step 5.

3.4.4 Numerical example

For production data shown in the previous section, determine an optimal group schedule minimizing the total tardiness. Fig. 3.5 shows the branching tree which was obtained with the use of the optimizing algorithm proposed.

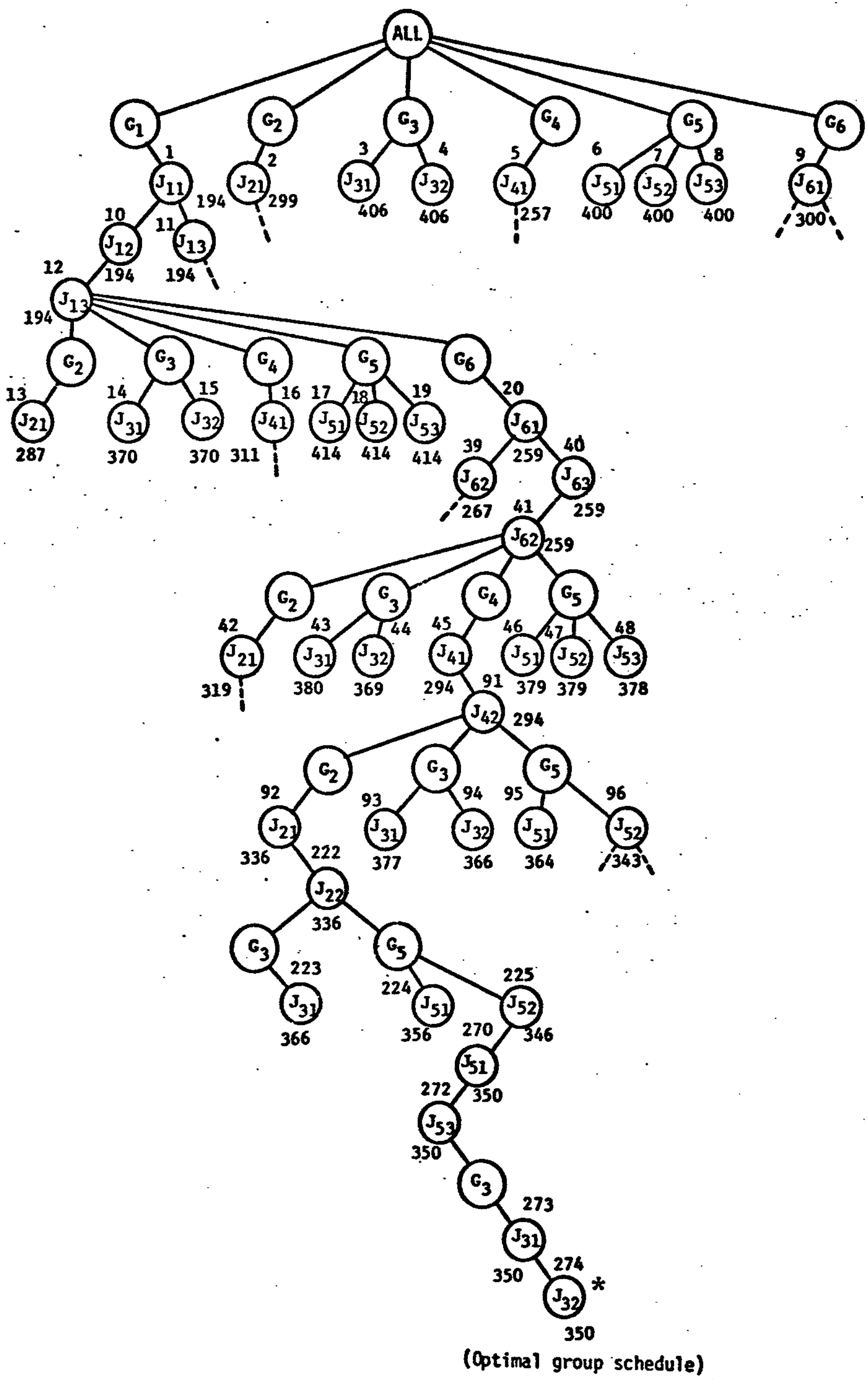


Fig. 3.5 The branching tree for the minimum-total-tardiness problem

The figures just below the job nodes indicate the lower bounds on the total tardiness. The order of branching is indicated by the number that appears just above the corresponding node. From this branching tree, an optimal group schedule is determined as $G_1(J_{11}-J_{12}-J_{13})-G_6(J_{61}-J_{63}-J_{62})-G_4(J_{41}-J_{42})$

$-G_2(J_{21}-J_{22})-G_5(J_{52}-J_{51}-J_{53})-G_3(J_{31}-J_{32})$ with the total tardiness of 350 min , which is given by the starred node. In this example, a complete enumeration for finding an optimal schedule will require 10368 times the comparisons of the feasible schedules, while the optimizing algorithm generated only 274 job nodes and required 10 seconds of CPU time with the TOSBAC model 140 computer to find an optimal one.

3.5 Conclusions

(1) The single-stage group scheduling model with sequence-dependent group setup times was developed and analyzed under three kinds of criteria — the minimum total elapsed time, the minimum mean flow time, and the minimum total tardiness.

(2) The minimum-total-elapsed-time problem was shown to be reduced to the traveling salesman problem.

(3) For the problem of minimizing the mean flow time, the optimal job sequence for each group was shown to be the SPT (shortest-processing-time) schedule, and the optimal group sequence was determined by applying the dynamic programming approach or the branch-and-bound method.

(3) The branch-and-bound method was applied to solve the problem with the objective of minimizing the total tardiness. Two theorems which specify the relative order of pairs of jobs in the same group were given. The optimizing algorithm which incorporated them as a part of the branching process was developed, and a numerical example was shown.

CHAPTER 4 GROUP SCHEDULING ON MULTIPLE PRODUCTION STAGES — THEORETICAL ANALYSIS

4.1 Introduction

The first step of development for scheduling theory on the multiple production stages dates back to Johnson's work for the two-stage (or two-machine) scheduling problem of minimizing the total elapsed time in 1954.[1] Johnson gave a theorem that establishes the relative order in which pairs of jobs are processed in an optimal schedule, and developed a working rule with which an optimal schedule can be easily constructed.

In general, for more than two-stage scheduling problems, no simple rules have been offered for determining the optimal schedule. However, Johnson showed in his original representation that a generalization of his theorem to the three-machine case is possible when the second machine is dominated. Moreover, Nabeshima,[2,3] Smith,[4] Gupta,[5] and Szwarc[6] solved the m-stage special structure flow-shop scheduling problems, where the processing times were not completely random but bore a well-defined relationship to one another.

On the other hand, Mitten,[7] Johnson,[8] and Nabeshima[9] considered the two-stage scheduling problems with time lags between the production of a job on the first machine and its production on the second one, and gave decision rules which are extensions of Johnson's theorem.

For the criteria except the minimization of the total elapsed time, the multistage scheduling problems have not been solved theoretically. As stated above, theories of scheduling on the multiple production stages are mainly concerned with the criterion of the minimum total elapsed time, and they have been developed based upon Johnson's theorem.

In general, when workpieces (parts) are processed on machines, setup times are needed to setup the machines for the processing of their operations. In the problems of a Johnson type, however, no attention has been directed to the setup times; that is, the setup times are assumed to be independent of the sequence and to be included in the processing times. In many actual problems, setup for a job and its processing happen to be independent of each other. Hence, setup for an operation of a job on a preceding machine can be done before completion of the operation of the job on the succeeding machine. In such a situation, it is not valid to absorb the setup time in the processing time. Therefore, from the standpoint of production scheduling, decisions as to the scheduling of jobs to be processed on more than two stages should be made by separating the setup times from the processing times.

Based on the above consideration, this chapter deals with the conventional scheduling and the group scheduling on the multiple production stages when the setup times are separated from the processing times.[10~12] First, Johnson's theorem for the two-stage scheduling problem is introduced, and then it is extended to the scheduling problem with setup times separated. In addition to the setup time consideration, the scheduling problems with time lags are dealt with in the latter part of Section 4.2.[13] In Section 4.3, the group scheduling problem with consideration of the setup times, which is the main objective in this chapter, is taken up for the two production stages. In the last section, this is extended to the group scheduling on the multiple production stages.

4.2 Two-stage scheduling problem with setup times separated and time lags

4.2.1 Two-stage scheduling problem of minimizing total elapsed time

Consider the scheduling problem which can be defined as follows: n jobs are given, each to be processed on two machines M_1 and M_2 in the same order. Given the processing time of each job on each machine, the problem is to find an operation schedule (job sequence) for each machine so as to minimize the total elapsed time.

For this problem, Johnson showed that it was sufficient to consider only schedules in which the same job sequence occurred on machines M_1 and M_2, and then proved the following well-known theorem. Let p_i^1 and p_i^2 ($i = 1, 2, \ldots, n$) denote the processing times of job J_i ($i = 1, 2, \ldots, n$) on machines M_1 and M_2, respectively.

Johnson's theorem An optimal ordering is given by the following rule: Job J_i precedes job J_j if

$$\min(p_i^1, p_j^2) < \min(p_j^1, p_i^2) \tag{4.1}$$

If there is equality, either ordering is optimal.

Based on this theorem, Johnson constructed a working rule for determining an optimal schedule.

In general, for the m-machine ($m \geq 3$) flow-shop scheduling problem of minimizing the total elapsed time, one needs to consider only schedules in which the same order is prescribed on the first two machines, and the same order is prescribed on the last two machines.[14)]

4.2.2 Scheduling problem with setup times separated

It is the purpose of this subsection to describe a scheduling model with setup times separated. For constructing a scheduling model of a new type, it is assumed that setup for an operation of a job on machine M_2 can be done before completion of the operation of the job on machine M_1 if there exist some idle times on machine M_2 (see Fig. 4.1). In the model, the time

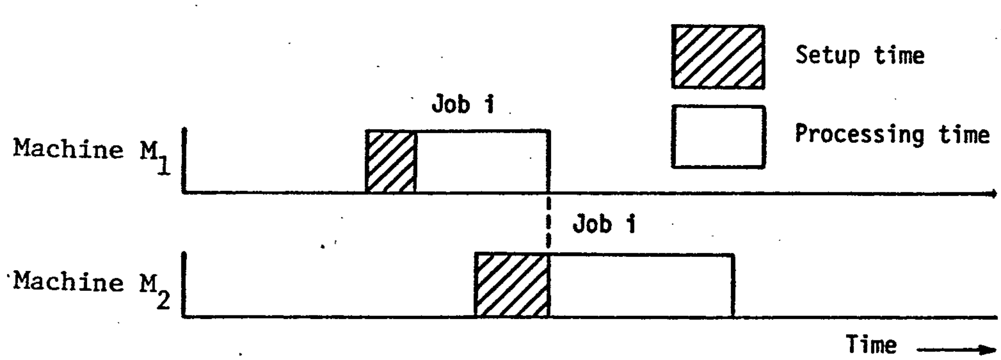


Fig. 4.1 Independence of setup and processing

required to complete each job on each machine consists of job setup time and job processing time, each of which is a scheduling unit.

First, examine whether the property that the same order on the first two machines and the last two machines will be sure to give an optimal schedule for the m-machine flow-shop scheduling is applicable to the problem with setup times separated. Even for the scheduling problem with setup times separated, the following theorem holds.

Theorem 4.1 For the flow-shop scheduling problem with setup times separated, it is sufficient to consider only schedules in which the same order occurs on the first two machines when the objective is to minimize the total elapsed time.

The proof of this theorem is omitted, since it can be easily proved with an argument which resembles that given for the proof for the problem with setup times included. It is worth noting that the separation of setup times from processing times makes it unnecessary for an optimal schedule to have the same order on the last two machines. A simple example will illustrate this. Suppose that two jobs are to be scheduled on a three-machine flow-shop to minimize the total elapsed time. The production data of the two jobs are shown in Table 4.1. There are two schedules S_1 and S_2 that

Table 4.1 Production data for three-machine scheduling

(units: hours)

Job	J_1		J_2	
Setup time· processing time	s_1^k	p_1^k	s_2^k	p_2^k
Machine M_1	1	3	1	1
Machine M_2	1	1	2	5
Machine M_3	5	1	1	4

have the same order on machines M_2 and M_3. The total elapsed times of the two schedules are 16 and 17 hours, respectively. However, there is a schedule S_3 with different orderings on machines M_2 and M_3 with the total elapsed time of 15 hours (see Fig. 4.2).

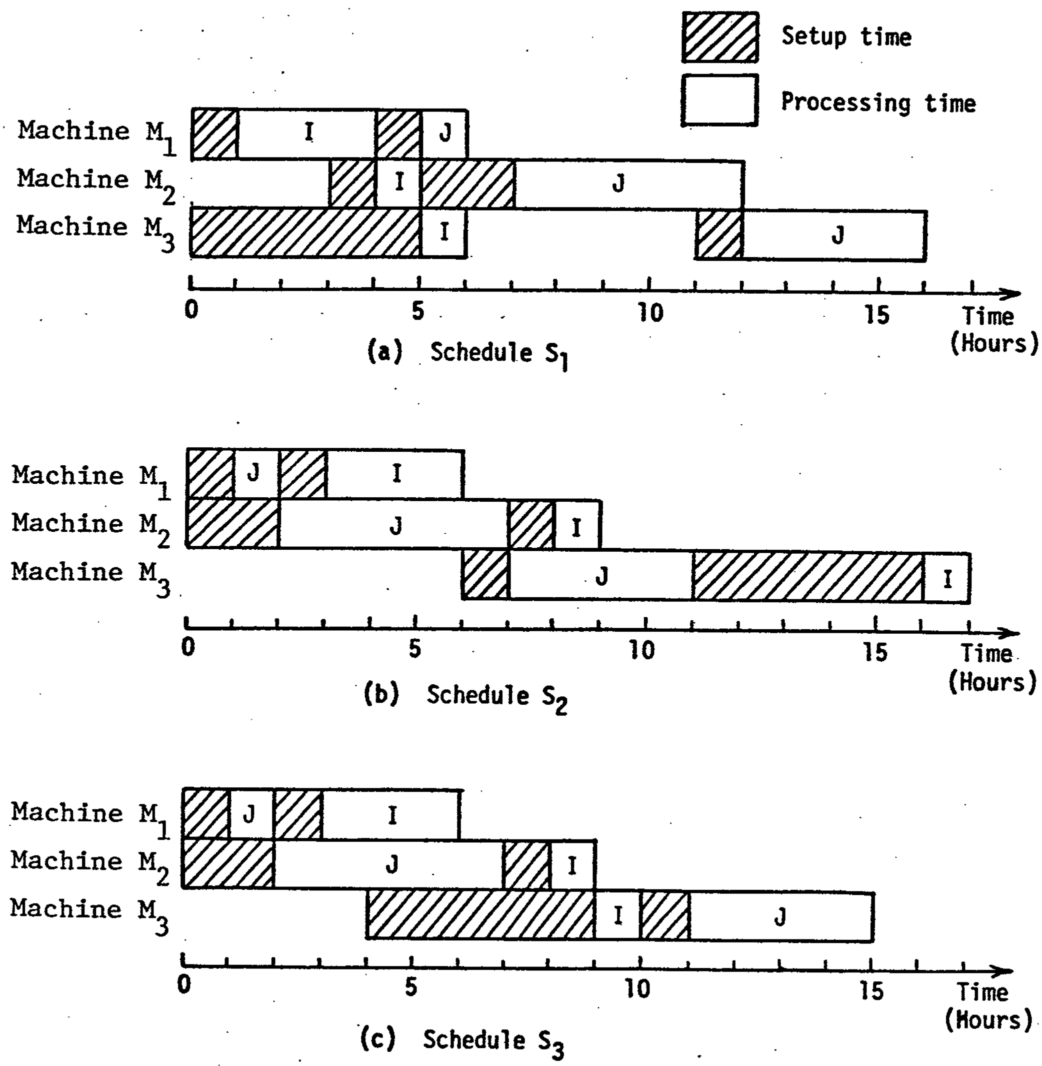


Fig. 4.2 Schedules on three machines

In the case of the two-machine flow-shop scheduling problem with setup times separated, the optimal schedule minimizing the total elapsed time can be characterized by the following rule for ordering pairs of jobs.

Theorem 4.2 For the two-stage flow-shop scheduling problem with setup times separated, an optimal schedule under the criterion of the minimum total elapsed time is given by the following rule:

Job J_i precedes job J_j if

$$\min(s_i^1 - s_i^2 + p_i^1,\ p_j^2) < \min(s_j^1 - s_j^2 + p_j^1,\ p_i^2) \tag{4.2}$$

If there is equality, either ordering is optimal.

Proof. The completion time of $J_{(i)}$ to be processed in the ith order on machine M_2 is given by the following recursive relation:

$$C^2_{(i)} = \max(C^1_{(i)},\ C^2_{(i-1)} + s^2_{(i)}) + p^2_{(i)} \tag{4.3}$$

where $C^1_{(i)} = \sum_{j=1}^{i}(s^1_{(j)} + p^1_{(j)})$ and $C^2_{(0)} = 0$.

By repeated use of relation (4.3), the total elapsed time is obtained as follows:

$$F_{max} = C^2_{(n)} = \max_{0\le u\le n}\{\sum_{i=1}^{u}(s^1_{(i)} + p^1_{(i)}) + p^2_{(u)} + \sum_{i=u+1}^{n}(s^2_{(i)} + p^2_{(i)})\}$$

$$= \max_{0\le u\le n}\{\sum_{i=1}^{u}(s^1_{(i)} - s^2_{(i)} + p^1_{(i)}) - \sum_{i=1}^{u-1}p^2_{(i)}\}$$

$$+ \sum_{i=1}^{n}(s^2_{(i)} + p^2_{(i)}) \tag{4.4}$$

where $p^2_{(0)} = 0$.

The problem is to find a sequence minimizing the above equation. Since the second summation is a constant, an equivalent problem is to minimize the first summation which indicates the idle times, I, on machine M_2. Hence, the objective function to be minimized is

$$I = \max_{0\le u\le n}\{\sum_{i=1}^{u}(s^1_{(i)} - s^2_{(i)} + p^1_{(i)}) - \sum_{i=1}^{u-1}p^2_{(i)}\} \tag{4.5}$$

By letting $r_i^1 = s_i^1 - s_i^2 + p_i^1$ and $r_i^2 = p_i^2$, equation (4.5) is

$$I = \max_{0 \le u \le n} \left(\sum_{i=1}^{u} r^1_{(i)} - \sum_{i=1}^{u-1} r^2_{(i)} \right) \tag{4.6}$$

Equation (4.6) is equivalent to the following:

$$I = \max\{0, \max_{1 \le u \le n} \left(\sum_{i=1}^{u} r^1_{(i)} - \sum_{i=1}^{u-1} r^2_{(i)} \right)\} \tag{4.7}$$

In general, it holds that max(0, A) $\le$ max(0, B) when A $\le$ B. Hence, the sequence which minimizes the value of $\max_{1 \le u \le n} (\sum_{i=1}^{u} r^1_{(i)} - \sum_{i=1}^{u-1} r^2_{(i)})$ also minimizes equation (4.7). Since the form of $\max_{1 \le u \le n} (\sum_{i=1}^{u} r^1_{(i)} - \sum_{i=1}^{u-1} r^2_{(i)})$ has the same one as that of Johnson's, an optimal ordering can be given by the following inequality:

$$\min(r^1_i, r^2_j) < \min(r^1_j, r^2_i) \tag{4.8}$$

(Q. E. D.)

This theorem is an extension of Johnson's, since by letting setup times be zero, inequality (4.2) becomes exactly the one Johnson gave in his paper. With an adaptation of this theorem, an optimal schedule is directly constructed by the following algorithm which is similar to Johnson's working rule.

< Optimizing algorithm for scheduling with setup times separated >

Step 1. Find the minimum value among the values of $(s^1_i - s^2_i + p^1_i)$ and p^2_i $(i = 1, 2, \ldots, n)$. (In the case of a tie, select arbitrarily.)

Step 2. If it is $(s^1_a - s^2_a + p^1_a)$, place J_a first, and if it is p^2_a, place J_a last.

Step 3. Remove the assigned job from consideration and go back to Step 1.

Consider a 4-job scheduling problem which has production data as shown in Table 4.2. By using the algorithm proposed, an optimal schedule is determined as J_2-J_4-J_1-J_3 with the total elapsed time of 41 hours. Table 4.3 shows the list of r^1_i and r^2_i for each job.

Table 4.2 Production data for two-machine scheduling

(units: hours)

Job	J_1		J_2		J_3		J_4	
Setup time, processing time	s_1^k	p_1^k	s_2^k	p_2^k	s_3^k	p_3^k	s_4^k	p_4^k
Machine M_1	2	10	1	5	2	7	3	8
Machine M_2	3	8	2	7	1	3	1	9

Table 4.3 List of r_i^1 and r_i^2

Job	J_1	J_2	J_3	J_4
r_i^1	9	4	8	10
r_i^2	8	7	3	9

In order to clarify the effect of the setup time consideration on the reduction of the total elapsed time, find an optimal schedule with setup times included. In this case, the processing times including the setup times are given by Table 4.4 and the optimal schedule is J_2-J_1-J_4-J_3 with the total elapsed time of 43 hours. Hence, the amount of the time reduction due to the setup time consideration is 2 hours for this example.

Table 4.4 List of processing times including setup times

Job	J_1	J_2	J_3	J_4
$s_i^1+p_i^1$	12	6	9	11
$s_i^2+p_i^2$	11	9	4	10

4.2.3 Two-stage scheduling problem with setup times separated and time lags

In the analysis of the previous subsection, no consideration was given to the transportation times between successive stages and the lap-phasing which may occur in lot production. In order to describe these situations, a model of another type is necessary.

Based on this consideration, a two-stage scheduling model with time lags has been constructed. In the model, two kinds of time lags (a start lag and a stop lag) are incorporated. The start lag (stop lag) prescribes that a job may not be started (completed) on the succeeding machine until at least a certain time has elapsed since starting (completing) the job on the preceding machine.

This problem is also an extension of Johnson's two-stage flow-shop problem, since Johnson's problem is one where the start lag and the stop lag for each job are set exactly equal to the job processing times on the preceding and succeeding machines, respectively. In the presented model, the use of different sequences on the two machines will yield a shorter elapsed time in some cases. In order to simplify the problem, it is assumed that the same order is to be used on both machines in the model.

In this subsection a further generalization is made to include the setup time consideration in the problem with time lags. The problem is to determine a schedule so as to minimize the total elapsed time when the setup times are separated from the processing times and jobs have their start lags and stop lags, respectively.

Let a_i and b_i be the start lag and the stop lag of job J_i, respectively (see Fig 4.3). In order to express the total elapsed time as a function of the processing time, the setup time, and the time lags of each job, develop the recursive function of the completion time of job $J_{(i)}$ to be processed in the ith order.

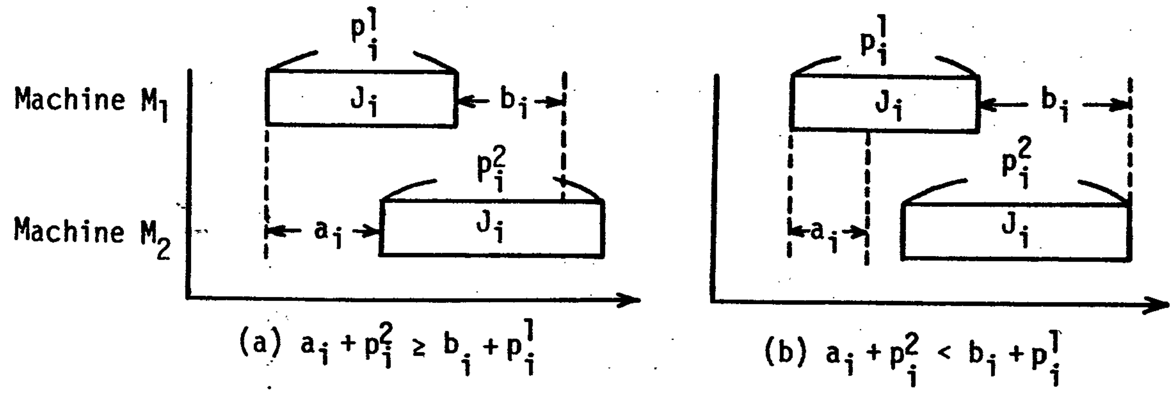


Fig. 4.3 Start lag a_i and stop lag b_i of J_i

In the case of start lag a_i and stop lag b_i, no job may be started on machine M_2 until at least a certain number of time units, T_i, which is given by the following equation, has elapsed since completion of the processing for the job on machine M_1.

$$T_i = \max(a_i,\ p_i^1 - p_i^2 + b_i) \tag{4.9}$$

Hence, the completion time of $J_{(i)}$ on machine M_2 is given by the following recursive relation (see Fig. 4.4):

$$C_{(i)}^2 = \max(C_{(i)}^1 - p_{(i)}^1 + T_{(i)},\ C_{(i-1)}^2 + s_{(i)}^2) + p_{(i)}^2 \tag{4.10}$$

where $C_{(i)}^1 = \sum_{j=1}^{i} (s_{(j)}^1 + p_{(j)}^1)$ and $C_{(0)}^2 = 0$.

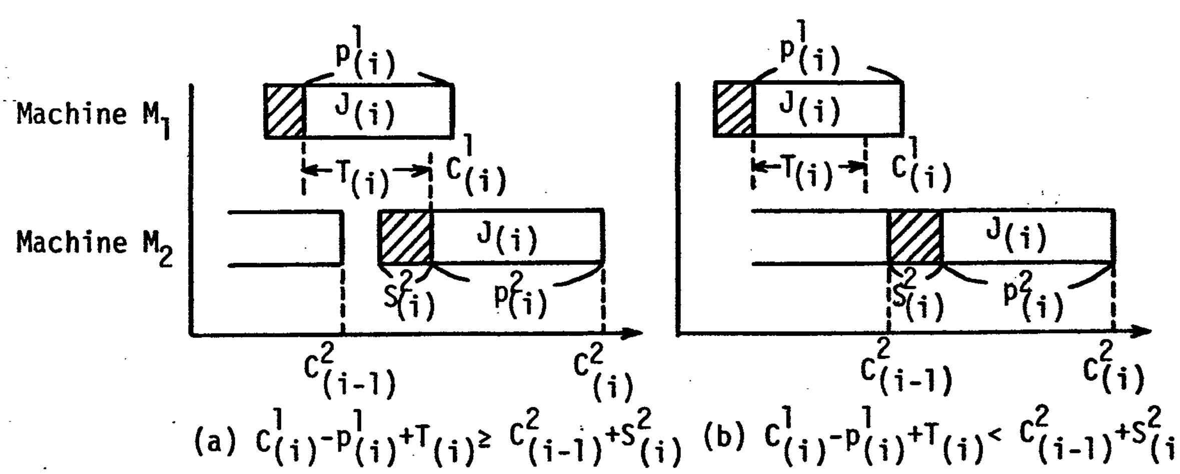


Fig. 4.4 Recursive relation of completion times of jobs

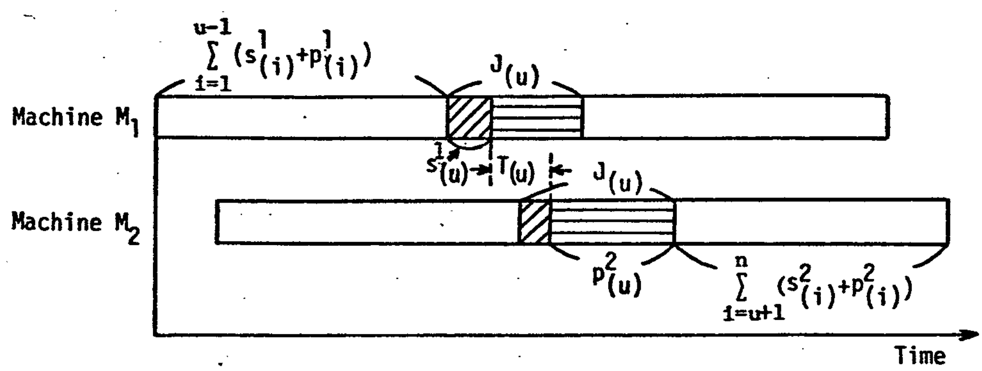
By repeatedly using the above relation, the total elapsed time is obtained by

$$F_{max} = C^2_{(n)} = \max_{0 \le u \le n} \{ \sum_{i=1}^{u-1} (s^1_{(i)} + p^2_{(i)}) + s^1_{(u)} + T_{(u)} + p^2_{(u)} + \sum_{i=u+1}^{n} (s^2_{(i)} + p^2_{(i)}) \}$$

$$= \max_{0 \le u \le n} \{ \sum_{i=1}^{u} (s^1_{(i)} - s^2_{(i)} + T_{(i)}) - \sum_{i=1}^{u-1} (p^2_{(i)} - p^1_{(i)} + T_{(i)}) \} + \sum_{i=1}^{n} (s^2_{(i)} + p^2_{(i)}) \qquad (4.11)$$

where $s^1_{(0)} = T_{(0)} = p^2_{(0)} = 0$.

The first expression of the above equation can be also obtained by considering the job $J_{(u)}$ to be critical in a schedule as shown in Fig. 4.5.



11

Fig. 4.5 Computation of the total elapsed time by assuming $J_{(u)}$ to be critical in a schedule

Since the second summation of equation (4.11) is a constant, the problem of minimizing the total elapsed time is equivalent to minimizing the first summation. This sum represents the total idle time I on machine M_2. Hence, the problem becomes one of finding a schedule so as to minimize the following equation:

$$I = \max_{0 \le u \le n} \{ \sum_{i=1}^{u} (s^1_{(i)} - s^2_{(i)} + T_{(i)}) - \sum_{i=1}^{u-1} (p^2_{(i)} - P^1_{(i)} + T_{(i)}) \} \qquad (4.12)$$

By letting $r_i^1 = s_i^1 - s_i^2 + T_i$ and $r_i^2 = p_i^2 - p_i^1 + T_i$, the above equation takes the form of $\max_{0 \le u \le n} (\sum_{i=1}^{u} r_{(i)}^1 - \sum_{i=1}^{u-1} r_{(i)}^2)$. This is the same form as the one in equation (4.6). Hence, with the same argument as in the case of the proof of Theorem 4.2, the following theorem can be derived for optimally sequencing the jobs.

Theorem 4.3 For the two-stage flow-shop scheduling problem with setup times separated and time lags, an optimal schedule under the criterion of the minimum total elapsed time is given by the following rule:

Job J_i precedes job J_j if

$$\min(s_i^1 - s_i^2 + T_i,\ p_j^2 - p_j^1 + T_j) < \min(s_j^1 - s_j^2 + T_j,\ p_i^2 - p_i^1 + T_i) \tag{4.13}$$

If there is equality, either ordering is optimal.

With the help of this theorem, an optimal schedule is directly determined by a similar algorithm to the one for the problem with setup times separated.

As a numerical example, consider a 4-job scheduling problem which has the same data as shown in Table 4.2, and which has other data of start and stop lags for each job as shown in Table 4.5.

Table 4.5 Start lags and stop lags

(units: hours)

Job	J_1	J_2	J_3	J_4
Start lag a_i	8	5	7	6
Stop lag b_i	5	4	8	4

The values of T_i for each job are computed as $(T_1, T_2, T_3, T_4) = (8, 5, 12, 6)$. For example, $T_1 = \max(8, 10-8+5) = 8$. Then the values of r_i^1 $(= s_i^1 - s_i^2 + T_i)$ and r_i^2 $(= p_i^2 - p_i^1 + T_i)$ for each job are calculated as $\{(r_1^1, r_1^2), (r_2^1, r_2^2), (r_3^1, r_3^2), (r_4^1, r_4^2)\} = \{(7, 6), (4, 7), (13, 8), (8, 7)\}$. Hence, the optimal schedule is determined as J_2-J_4-J_3-J_1.

4.3 Extension of two-stage scheduling to group scheduling

An analysis of the previous section is made to the conventional scheduling problem in which there is only one group consisting of n jobs to be processed. In this section, the two-stage flow-shop group scheduling problem will be theoretically treated. In the group scheduling to be analyzed, hereafter, no attention will be paid to the time lags of jobs. However, it is possible to incorporate the time lag consideration into the group scheduling model.

The two-stage string problem, which is a kind of group scheduling, has been studied by Kurisu.[15, 16] In the string problem, it is assumed that the order of jobs within each group (string) is fixed. That is, an optimal decision is made only as to the sequence of groups classified. In the model, group setup times which will play an important role in grouping jobs are not considered.

In the two-stage flow-shop group scheduling for the minimum total elapsed time, the following theorem can be proved in the same manner as in the lemma by Johnson.

Theorem 4.4 For the two-stage flow-shop group scheduling problem of minimizing the total elapsed time, it is sufficient to consider only group schedules in which the same orders of groups and jobs occur on both machines.

Let $p^k_{i\xi}$ ($i=1, 2,\ldots, N$, $\xi=1, 2,\ldots, n_i$, $k=1, 2,\ldots, K$) denote the job processing time including the job setup time on stage (machine) M_k ($k=1, 2,\ldots, K$) of job $J_{i\xi}$ ($i=1, 2,\ldots, N$, $\xi=1, 2,\ldots, n_i$) of group G_i ($i=1, 2, \ldots, N$) and S^k_i ($i=1, 2,\ldots, N$, $k=1, 2,\ldots, K$) denote the group setup time on stage M_k of group G_i. For the sake of convenience, the job processing time and the group setup time are defined on stage M_k ($k=1, 2,\ldots, K$), respectively.

First, develop the recursive relation of the completion time on machine M_k of job $J_{(i)(\xi)}$, which indicates the ξth job in the ith group in a group schedule. The completion time of $J_{(i)(\xi)}$ on M_k is given by

$$\left.\begin{aligned} C^k_{(i)(\xi)} &= \max(C^{k-1}_{(i)(\xi)}, C^k_{(i)(\xi-1)}) + p^k_{(i)(\xi)}, \quad \text{if } \xi \neq 1 \\ C^k_{(i)(1)} &= \max(C^{k-1}_{(i)(1)}, C^k_{(i-1)(n_{i-1})} + S^k_{(i)}) + p^k_{(i)(1)}, \quad \text{if } \xi = 1 \end{aligned}\right\} \tag{4.14}$$

where $C^0_{(i)(\xi)} = C^k_{(0)(n_0)} = 0$.

By using these recursive relations repeatedly, the total elapsed time for the two-stage flow-shop problem, the completion time on M_2 of $J_{(N)(n_N)}$, is expressed as

$$\begin{aligned} F_{max} = C^2_{(N)(n_N)} &= \max_{0\le u\le N} \max_{1\le \nu\le n_u} \{\sum_{i=1}^{u-1} (S^1_{(i)} + \sum_{\xi=1}^{n_i} p^1_{(i)(\xi)}) + S^1_{(u)} \\ &\quad + \sum_{\xi=1}^{\nu} p^1_{(u)(\xi)} + \sum_{\xi=\nu}^{n_u} p^2_{(u)(\xi)} + \sum_{i=u+1}^{N} (S^2_{(i)} + \sum_{\xi=1}^{n_i} p^2_{(i)(\xi)})\} \\ &= \max_{0\le u\le N} \{\sum_{i=1}^{u-1} (S^1_{(i)} + P^1_{(i)}) + S^1_{(u)} + \max_{1\le \nu\le n_u} (\sum_{\xi=1}^{\nu} p^1_{(u)(\xi)} + \sum_{\xi=\nu}^{n_u} p^2_{(u)(\xi)}) \\ &\quad + \sum_{i=u+1}^{N} (S^2_{(i)} + P^2_{(i)})\} \end{aligned} \tag{4.15}$$

where $P^k_{(i)} = \sum_{\xi=1}^{n_i} p^k_{(i)(\xi)}$, $p^k_{(0)(\xi)} = 0$ $(k = 1, 2)$, and $S^1_{(0)} = 0$, and $n_{(0)} \geq 1$.

The problem is to determine a group schedule so as to minimize the above equation. The following theorem holds for this problem.

Theorem 4.5 For the two-stage flow-shop group scheduling problem for the minimum total elapsed time, an optimal group schedule is obtained by the following rules; the job sequence is by Rule 1, and the group sequence by Rule 2.

Rule 1: Job $J_{i\xi}$ precedes job $J_{i\eta}$ if

$$\min(p^1_{i\xi}, p^2_{i\eta}) < \min(p^1_{i\eta}, p^2_{i\xi}) \tag{4.16}$$

Rule 2: Group G_i precedes group G_j if

$$\min \{S^1_i - S^2_i + \max_{1\le\nu\le n_i} (\sum_{\xi=1}^{\nu} p^1_{i\xi} - \sum_{\xi=1}^{\nu-1} p^2_{i\xi}), \max_{1\le\nu\le n_j} (\sum_{\xi=\nu}^{n_j} p^2_{j\xi} - \sum_{\xi=\nu+1}^{n_j} p^1_{j\xi})\}$$

$$< \min \{S^1_j - S^2_j + \max_{1\le\nu\le n_j} (\sum_{\xi=1}^{\nu} p^1_{j\xi} - \sum_{\xi=1}^{\nu-1} p^2_{j\xi}),$$

$$\max_{1\le\nu\le n_i} (\sum_{\xi=\nu}^{n_i} p^2_{i\xi} - \sum_{\xi=\nu+1}^{n_i} p^1_{i\xi})\} \tag{4.17}$$

If there is equality in inequality (4.16) or (4.17), either ordering is optimal for group and job sequences, respectively.

Proof: The optimal schedule under the criterion of the minimum total elapsed time can be obtained by minimizing the total idle time at the second stage. From equation (4.15), the total idle time at this stage is given by

$$I = F_{max} - \sum_{i=1}^{N} (S^2_{(i)} + P^2_{(i)})$$

$$= \max_{0\le u\le N} \max_{1\le\nu\le n_u} \{\sum_{i=1}^{u-1} (S^1_{(i)} + P^1_{(i)} - S^2_{(i)} - P^2_{(i)}) + S^1_{(u)} - S^2_{(u)}$$

$$+ \sum_{\xi=1}^{\nu} p^1_{(u)(\xi)} - \sum_{\xi=1}^{\nu-1} p^2_{(u)(\xi)}\} \tag{4.18}$$

This is equivalent to the following:

$$I = \max_{0\le u\le N} \{\sum_{i=1}^{u-1} (S^1_{(i)} + P^1_{(i)} - S^2_{(i)} - P^2_{(i)}) + S^1_{(u)} - S^2_{(u)}$$

$$+ \max_{1\le\nu\le n_u} (\sum_{\xi=1}^{\nu} p^1_{(u)(\xi)} - \sum_{\xi=1}^{\nu-1} p^2_{(u)(\xi)})\}$$

$$= \max_{0\le u\le N} \{ \sum_{i=1}^{u-1} (Q^1_{(i)} - Q^2_{(i)}) + S^1_{(u)} - S^2_{(u)}$$

$$+ \max_{1\le\nu\le n_u} (\sum_{\xi=1}^{\nu} p^1_{(u)(\xi)} - \sum_{\xi=1}^{\nu-1} p^2_{(u)(\xi)}) \} \qquad (4.19)$$

where $Q^k_{(i)} = S^k_{(i)} + P^k_{(i)}$ $(k = 1,\ 2)$.

Hence, for any group sequence, the total idle time at the second stage is minimized by determining the job sequence within each group $G_{(u)}$ so as to minimize $\max_{1\le\nu\le n_u} (\sum_{\xi=1}^{\nu} p^1_{(u)(\xi)} - \sum_{\xi=1}^{\nu-1} p^2_{(u)(\xi)})$. This is accomplished by ordering the jobs using Rule 1.

Introducing $R_{(u)} = S^1_{(u)} - S^2_{(u)}$ and $V_{(u)} = \max_{1\le\nu\le n_u} (\sum_{\xi=1}^{\nu} p^1_{(u)(\xi)} - \sum_{\xi=1}^{\nu-1} p^2_{(u)(\xi)})$, equation (4.19) is presented by

$$I = \max_{0\le u\le N} \{ \sum_{i=1}^{u-1} (Q^1_{(i)} - Q^2_{(i)}) + R_{(u)} + V_{(u)} \} \qquad (4.20)$$

Rule 2 as to the determination of an optimal group sequence is proved in the following way. Let w be any group sequence and I(w) be its total idle time at stage M_2. Consider a group sequence w' that differs from w only in that two consecutive groups $G_{(i)}$ and $G_{(i+1)}$ are interchanged in w.

Develop a sufficient condition such that I(w) < I(w'). Inequality I(w) < I(w') is equivalent to the following:

$$\max\{ \sum_{j=1}^{i-1} (Q^1_{(j)} - Q^2_{(j)}) + R_{(i)} + V_{(i)},\ \sum_{j=1}^{i} (Q^1_{(j)} - Q^2_{(j)}) + R_{(i+1)} + V_{(i+1)} \}$$

$$< \max\{ \sum_{j=1}^{i-1} (Q^1_{(j)} - Q^2_{(j)}) + R_{(i+1)} + V_{(i+1)},$$

$$\sum_{j=1}^{i-1} (Q^1_{(j)} - Q^2_{(j)}) + Q^1_{(i+1)} - Q^2_{(i+1)} + R_{(i)} + V_{(i)} \} \qquad (4.21)$$

By subtracting $\sum_{j=1}^{i-1} (Q^1_{(j)} - Q^2_{(j)}) + R_{(i)} + V_{(i)} + R_{(i+1)} + V_{(i+1)}$ from each term of inequality (4.21), the following one is obtained:

$$\max(-R_{(i+1)} - V_{(i+1)},\ Q^1_{(i)} - Q^2_{(i)} - R_{(i)} - V_{(i)})$$

$$< \max(-R_{(i)} - V_{(i)},\ Q^1_{(i+1)} - Q^2_{(i+1)} - R_{(i+1)} - V_{(i+1)}) \tag{4.22}$$

Since $\max(x, y) = -\min(-x, -y)$, this is equivalent to the following:

$$\min(R_{(i+1)} + V_{(i+1)},\ R_{(i)} + V_{(i)} - Q^1_{(i)} - Q^2_{(i)})$$

$$< \min(R_{(i)} + V_{(i)},\ R_{(i+1)} + V_{(i+1)} - Q^1_{(i+1)} - Q^2_{(i+1)}) \tag{4.23}$$

upon this, each term in the parentheses of the above inequality is transformed into the following one:

$$R_{(i)} + V_{(i)} = S^1_{(i)} - S^2_{(i)} + \max_{1 \le \nu \le n_i} \left(\sum_{\xi=1}^{\nu} p^1_{(i)(\xi)} - \sum_{\xi=1}^{\nu-1} p^2_{(i)(\xi)} \right) \tag{4.24}$$

$$R_{(i)} + V_{(i)} - Q^1_{(i)} - Q^2_{(i)} = \max_{1 \le \nu \le n_i} \left(\sum_{\xi=\nu}^{n_i} p^2_{(i)(\xi)} - \sum_{\xi=\nu+1}^{n_i} p^1_{(i)(\xi)} \right) \tag{4.25}$$

From these expressions (4.23), (4.24), and (4.25), and the fact that the ordering of the two consecutive groups given by inequality (4.23) is transitive, it follows that Rule 2 characterizes an optimal group sequence.

(Q. E. D.)

This theorem is an extension of Johnson's to group scheduling. With the use of this theorem, a simple algorithm for determining an optimal group schedule is developed as follows:

< Optimizing algorithm for the two-stage flow-shop group scheduling >

Step 1. Determine an optimal job sequence in each group by using Johnson's working rule.

Step 2. Determine an optimal group sequence in the following way:

(i) Calculate the following values for each group under the job sequences determined by Step 1.

$$X_i = S_i^1 - S_i^2 + \max_{1 \le \nu \le n_i} \left(\sum_{\xi=1}^{\nu} p_{i\xi}^1 - \sum_{\xi=1}^{\nu-1} p_{i\xi}^2 \right)$$

$$Y_i = \max_{1 \le \nu \le n_i} \left(\sum_{\xi=\nu}^{n_i} p_{i\xi}^2 - \sum_{\xi=\nu+1}^{n_i} p_{i\xi}^1 \right)$$

(ii) Find the minimum value among the X_i's and the Y_i's. (In the case of a tie, select arbitrarily.)

(iii) If it is X_a, place G_a first, and if it is Y_a, place G_a last.

(iv) Remove the assigned group from consideration and go back to (ii).

4.4 Optimal group scheduling on multiple production stages

The recent advances in scheduling technique have shown that it is rather difficult to develop simple optimizing algorithms for solving the general flow-shop scheduling problem with the simple criterion of minimizing the total elapsed time, much less the problems with more complex measures, such as minimizing the mean flow time. As a result of this awareness, a direction of recent research in multistage scheduling problems has been turned to the special structure scheduling problems which can be easily solved theoretically. Thus, several cases in which the job processing times bear well-defined relationships have been considered, and efficient optimizing algorithms for determining the optimal schedules have been developed.

In this section, these special cases in the conventional scheduling are generalized to the group scheduling. A theoretical determination of the optimal group schedule under the criterion of the minimum total elapsed time can be made to the special structure flow-shop scheduling problems where there exist some well-defined relationships among the group processing times and the job processing times.

For the K-stage ($K \ge 3$) flow-shop group scheduling problem, the schedules having the same order of groups and jobs on the first two machines include an optimal schedule under the criterion of minimizing the total elapsed time, while the optimal schedule does not always have the same order on the last two machines. The latter property does not occur because of the grouping of jobs into several groups but because of the feature of the group setup time.

In the K-stage flow-shop group scheduling problem which will be treated hereafter, it is assumed that the processing order of groups and jobs is the same on each machine. (No passing of groups and jobs is allowed.)

With the aid of equation (4.14), the total elapsed time for the K-stage flow-shop problem is given by

$$F_{max} = C^K_{(N)(n_N)} = \max_{0 \le u_1 \le u_2 \le \cdots \le u_{K-1} \le N} \quad \max_{\substack{\nu_k \in \Gamma_k \\ (k=1,2,\ldots,K-1)}} \sum_{k=1}^{K} R(u_{k-1}, u_k) \tag{4.26}$$

where

(i) for $k=1$, $\quad R(u_0, u_1) = \sum_{i=1}^{u_1-1} Q^1_{(i)} + S^1_{(u_1)} + \sum_{\xi=1}^{\nu_1} p^1_{(u_1)(\xi)}$

(ii) for $1 < k < K-1$

$$R(u_{k-1}, u_k) = \begin{cases} \sum_{\xi=\nu_{k-1}}^{n_{u_{k-1}}} p^k_{(u_{k-1})(\xi)} + \sum_{i=u_{k-1}+1}^{u_k-1} Q^k_{(i)} + S^k_{(u_k)} + \sum_{\xi=1}^{\nu_k} p^k_{(u_k)(\xi)} & (u_{k-1} < u_k) \\ \sum_{\xi=\nu_{k-1}}^{\nu_k} p^k_{(u_k)(\xi)} & (u_{k-1} = u_k) \end{cases}$$

(iii) for $k=K$,

$$R(u_{K-1}, u_K) = \sum_{\xi=\nu_{K-1}}^{n_{u_{K-1}}} p^K_{(u_{K-1})(\xi)} + \sum_{i=u_{K-1}+1}^{N} Q^K_{(i)}$$

and $S^k_{(0)} = 0$, $p^k_{(0)(\xi)} = 0$, and Γ_k is a set of ν_k such that

(i) $1 \le \nu_k \le n_{u_k} \quad (u_{k-1} < u_k < u_{k+1})$

(ii) $1 \le \nu_k \le \nu_{k+1} \quad (u_{k-1} < u_k = u_{k+1})$

(iii) $\nu_{k-1} \le \nu_k \le n_{u_k} \quad (u_{k-1} = u_k < u_{k+1})$

(iv) $\nu_{k-1} \le \nu_k \le \nu_{k+1} \quad (u_{k-1} = u_k = u_{k+1})$

where $u_0 = 0$, $u_K = N$, $\nu_0 = 1$, $\nu_K = n_N$, and $n_0 \ge 1$.

4.4.1 Optimal group scheduling for three special cases

It is obvious from equation (4.26) that the task of determining a schedule so as to minimize it is more formidable. However, if each of the following well-defined relationships holds among the group setup times and the job processing times at each of the stages, the problem can be reduced to a two-stage one, and hence solved theoretically.

Case 1: For a fixed $h \le (K-1)$, the group setup times and the job processing times satisfy the following conditions:

$$\left.\begin{array}{l}\left.\begin{array}{l}\text{(i)}\ \min\limits_{1\le i\le N}(S_i^k - S_i^{k+1} + \min\limits_{1\le\xi\le n_i} p_{i\xi}^k) \ge \max\limits_{1\le j\le N}\ \max\limits_{1\le\eta\le n_j} p_{j\eta}^{k+1} \\ \quad \min\limits_{1\le\xi\le n_i} p_{i\xi}^k \le \max\limits_{1\le\eta\le n_i} p_{i\eta}^{k+1} \quad (i = 1, 2, \ldots, N)\end{array}\right\} \forall k \le h-1 \le K-2 \\ \left.\begin{array}{l}\text{(ii)}\ \max\limits_{1\le i\le N}(S_i^k - S_i^{k+1} + \max\limits_{1\le\xi\le n_i} p_{i\xi}^k) \le \min\limits_{1\le j\le N}\ \min\limits_{1\le\eta\le n_j} p_{j\eta}^{k+1} \\ \quad \max\limits_{1\le\xi\le n_i} p_{i\xi}^k \le \min\limits_{1\le\eta\le n_i} p_{i\eta}^{k+1} \quad (i = 1, 2, \ldots, N)\end{array}\right\} h+1 \le \forall k \le K-1\end{array}\right\} \quad (4.27)$$

Case 2: The group setup times and the job processing times satisfy the following conditions:

$$\left.\begin{array}{l}\max\limits_{1\le i\le N}(S_i^k - S_i^{k+1} + \max\limits_{1\le\xi\le n_i} p_{i\xi}^k) \le \min\limits_{1\le j\le N}\ \min\limits_{1\le\eta\le n_j} p_{j\eta}^{k+1} \\ \max\limits_{1\le\xi\le n_i} p_{i\xi}^k \le \min\limits_{1\le\eta\le n_i} p_{i\eta}^{k+1} \quad (i = 1, 2, \ldots, N)\end{array}\right\} \forall k \le K-2 \quad (4.28)$$

Case 3: The group setup times and the job processing times satisfy the following conditions:

$$\left.\begin{aligned}&\min_{1\le i\le N}\left(S_i^k - S_i^{k+1} + \min_{1\le\xi\le n_i} p_{i\xi}^k\right) \ge \max_{1\le j\le N}\ \max_{1\le\eta\le n_j} p_{j\eta}^{k+1}\\ &\min_{1\le\xi\le n_i} p_{i\xi}^k \ge \max_{1\le\eta\le n_i} p_{i\eta}^{k+1} \qquad (i=1,\ 2,\ldots,\ N)\end{aligned}\right\}\ 2\le\forall k\le K-1 \tag{4.29}$$

Note that none of the conditions above is required when $K=2$. The following theorem holds for Case 1.

Theorem 4.6 If the group setup times and the job processing times satisfy conditions (4.27), then an optimal group schedule is obtained by determining the job sequence for each group using Rule 1, and the group sequence using Rule 2.

Rule 1: Job $J_{i\xi}$ precedes job $J_{i\eta}$ if

$$\min\left(\sum_{k=1}^{K-1} p_{i\xi}^k,\ \sum_{k=2}^{K} p_{i\eta}^k\right) < \min\left(\sum_{k=1}^{K-1} p_{i\eta}^k,\ \sum_{k=2}^{K} p_{i\xi}^k\right) \tag{4.30}$$

Rule 2: Group G_i precedes group G_j if

$$\begin{aligned}&\min\Bigl\{\sum_{k=1}^{K-1} S_i^k - \sum_{k=2}^{K} S_i^k + \max_{1\le\nu\le n_i}\Bigl(\sum_{\xi=1}^{\nu}\sum_{k=1}^{K-1} p_{i\xi}^k - \sum_{\xi=1}^{\nu-1}\sum_{k=2}^{K} p_{i\xi}^k\Bigr),\\ &\qquad \max_{1\le\nu\le n_j}\Bigl(\sum_{\xi=\nu}^{n_j}\sum_{k=2}^{K} p_{j\xi}^k - \sum_{\xi=\nu+1}^{n_j}\sum_{k=1}^{K-1} p_{j\xi}^k\Bigr)\Bigr\}\\ &< \min\Bigl\{\sum_{k=1}^{K-1} S_j^k - \sum_{k=2}^{K} S_j^k + \max_{1\le\nu\le n_j}\Bigl(\sum_{\xi=1}^{\nu}\sum_{k=1}^{K-1} p_{j\xi}^k - \sum_{\xi=1}^{\nu-1}\sum_{k=2}^{K} p_{j\xi}^k\Bigr),\\ &\qquad \max_{1\le\nu\le n_i}\Bigl(\sum_{\xi=\nu}^{n_i}\sum_{k=2}^{K} p_{i\xi}^k - \sum_{\xi=\nu+1}^{n_i}\sum_{k=1}^{K-1} p_{i\xi}^k\Bigr)\Bigr\}\end{aligned} \tag{4.31}$$

If there is equality in inequality (4.30) or (4.31), either ordering is optimal for group and job sequences, respectively.

Proof: The problem is to find a group schedule such that the total elapsed time given by equation (4.26) is minimized. By subtracting $\sum_{i=1}^{N}(S_{(i)}^{K}+P_{(i)}^{K})$ from the equation and arranging it, the total idle time at the last stage is obtained as follows:

$$I = \max_{0\le u_1\le u_2\le\cdots\le u_{K-1}\le N} \quad \max_{\substack{\nu_k\in\Gamma_k\\(k=1,2,\ldots,K-1)}} \Big[\sum_{k=1}^{K-1}\Big\{\sum_{i=1}^{u_k-1}(Q_{(i)}^{k}-Q_{(i)}^{k+1})$$

$$+ S_{(u_k)}^{k} - S_{(u_k)}^{k+1} + \sum_{\xi=1}^{\nu_k} P_{(u_k)(\xi)}^{k} - \sum_{\xi=1}^{\nu_k-1} P_{(u_k)(\xi)}^{k+1}\Big\}\Big] \tag{4.32}$$

Since the value of $\sum_{i=1}^{N}(S_{(i)}^{K}+P_{(i)}^{K})$ is a constant, the problem of minimizing equation (4.26) is equivalent to minimizing the above equation.

Introduce

$$H_{u\nu}^{k} = \sum_{i=1}^{u-1}(Q_i^k - Q_i^{k+1}) + S_u^k - S_u^{k+1} + \sum_{\xi=1}^{\nu_u} p_{u\xi}^{k} - \sum_{\xi=1}^{\nu_u-1} p_{u\xi}^{k+1} \tag{4.33}$$

From the obvious identities

$$\left.\begin{aligned} H_{u\nu+1}^{k} &= H_{u\nu}^{k} + p_{u\nu+1}^{k} - p_{u\nu}^{k+1} \\ H_{u1}^{k} &= H_{u-1n_{u-1}}^{k} + S_u^k - S_u^{k+1} + p_{u1}^{k} - p_{u-1n_{u-1}}^{k+1} \end{aligned}\right\} \tag{4.34}$$

and conditions (4.27), the following inequalities hold

$$\left.\begin{aligned} &H_{u\nu}^{k} \le H_{u\nu+1}^{k} \quad\text{and}\quad H_{u-1n_{u-1}}^{k} \le H_{u1}^{k} \quad (k=1,\ 2,\ldots,\ h-1) \\ &H_{u\nu}^{k} \ge H_{u\nu+1}^{k} \quad\text{and}\quad H_{u-1n_{u-1}}^{k} \ge H_{u1}^{k} \quad (k=h+1,\ h+2,\ldots,\ K-1) \end{aligned}\right\} \tag{4.35}$$

Hence, equation (4.32) can be presented in the following form:

$$I = \max_{0\le u_h\le N}\ \max_{1\le\nu_h\le n_{u_h}} \sum_{k=1}^{K-1} H_{(u_h)(\nu_h)}^{k} \tag{4.36}$$

Denote u_h and ν_h by u and ν, respectively.

Then

$$I = \max_{0\le u\le N} \max_{1\le \nu\le n_u} [\sum_{k=1}^{K-1} \{ \sum_{i=1}^{u-1} (Q^k_{(i)} - Q^{k+1}_{(i)}) + S^k_{(u)} - S^{k+1}_{(u)}$$

$$+ \sum_{\xi=1}^{\nu} p^k_{(u)(\xi)} - \sum_{\xi=1}^{\nu-1} p^{k+1}_{(u)(\xi)} \}]$$

$$= \max_{0\le u\le N} \{ \sum_{i=1}^{u-1} (\sum_{k=1}^{K-1} Q^k_{(i)} - \sum_{k=2}^{K} Q^k_{(i)}) + \sum_{k=1}^{K-1} S^k_{(u)} - \sum_{k=2}^{K} S^k_{(u)} \}$$

$$+ \max_{1\le \nu\le n_u} (\sum_{\xi=1}^{\nu}\sum_{k=1}^{K-1} p^k_{(u)(\xi)} - \sum_{\xi=1}^{\nu-1}\sum_{k=2}^{K} p^k_{(u)(\xi)}) \tag{4.37}$$

Let $r^1_{i\xi} = \sum_{k=1}^{K-1} p^k_{i\xi}$, $r^2_{i\xi} = \sum_{k=2}^{K} p^k_{i\xi}$, $W^1_i = \sum_{k=1}^{K-1} S^k_i$, $W^2_i = \sum_{k=2}^{K} S^k_i$, $U^1_i = W^1_i + \sum_{\xi=1}^{n_i} r^1_{i\xi}$, and $U^2_i = W^2_i + \sum_{\xi=1}^{n_i} r^2_{i\xi}$, then $U^1_i = \sum_{k=1}^{K-1} S^k_i + \sum_{\xi=1}^{n_i}\sum_{k=1}^{K-1} p^k_{i\xi} = \sum_{k=1}^{K-1} Q^k_i$ and $U^2_i = \sum_{k=2}^{K} Q^k_i$.

Hence, the total idle time at stage M_K can be denoted by

$$I = \max_{0\le u\le N} \{ \sum_{i=1}^{u-1} (U^1_{(i)} - U^2_{(i)}) + W^1_{(u)} - W^2_{(u)}$$

$$+ \max_{1\le \nu\le n_u} (\sum_{\xi=1}^{\nu} r^1_{(u)(\xi)} - \sum_{\xi=1}^{\nu-1} r^2_{(u)(\xi)})\} \tag{4.38}$$

This expression has the same form as equation (4.19). Therefore, replacing W^k_i and $r^k_{i\xi}$ $(k=1, 2)$ by S^k_i and $p^k_{i\xi}$ $(k=1, 2)$, respectively, in conditions (4.16) and (4.17) of Theorem 4.5, we get conditions (4.30) and (4.31).

(Q. E. D.)

With the help of this theorem, an optimal group schedule can be determined easily by the following algorithm similar to the one for the two-stage problem, since Rules 1 and 2 have the same forms as the rules of Theorem 4.5.

< Optimizing algorithm for case 1 >

Step 1. (Determining the optimal job sequences)

(i) Calculate the fictitious processing times, the values of $\sum_{k=1}^{K-1} p_{i\xi}^{k}$ $(= x_{i\xi})$ and $\sum_{k=2}^{K} p_{i\xi}^{k}$ $(= y_{i\xi})$ for each job.

(ii) Determine an optimal job sequence for each group by applying Johnson's working rule to these values.

Step 2. (Determining the optimal group sequence)

(i) Calculate the following values for each group under the job sequences determined by Step 1.

$$X_i = \sum_{k=1}^{K-1} S_i^k - \sum_{k=2}^{K} S_i^k + \max_{1 \le \nu \le n_i} \left(\sum_{\xi=1}^{\nu} \sum_{k=1}^{K-1} p_{i\xi}^k - \sum_{\xi=1}^{\nu-1} \sum_{k=2}^{K} p_{i\xi} \right)$$

$$Y_i = \max_{1 \le \nu \le n_i} \left(\sum_{\xi=\nu}^{n_i} \sum_{k=2}^{K} p_{i\xi}^k - \sum_{\xi=\nu+1}^{n_i} \sum_{k=1}^{K-1} p_{i\xi}^k \right)$$

(ii) Find the minimum value among the X_i's and the Y_i's. (In the case of a tie, select arbitrarily.)

(iii) If it is X_a, place G_a first, and if it is Y_a, place G_a last.

(iv) Remove the assigned group from consideration and go back to (ii).

When the group setup times and the job processing times satisfy conditions (4.28) of Case 2, it follows from equation (4.33) that

$$H_{u\nu}^k \ge H_{u\nu+1}^k \quad \text{and} \quad H_{u-1n_{u-1}}^k \ge H_{u1}^k \quad (k = 1, 2, \ldots, K-2) \tag{4.39}$$

Then

$$H_{(1)(1)}^k \ge H_{(i)(\xi)}^k \quad (k = 1, 2, \ldots, K-2) \tag{4.40}$$

Hence, the total idle time at the last stage given by equation (4.32) can be denoted as follows:

$$I = \max_{\substack{0 \le u_1 \le u_2 \le \cdots \le u_{K-2} \le 1 \\ u_{K-2} \le u_{K-1} \le N}} \ \max_{\substack{\nu_k = 1 \\ (k=1,2,\ldots,K-2) \\ 1 \le \nu_{K-1} \le n_{u_{K-1}}}} \left(\sum_{k=1}^{K-2} H_{(u_k)(\nu_k)}^k + H_{(u_{K-1})(\nu_{K-1})}^{K-1} \right)$$

$$= \max_{0 \le u_1 \le u_2 \le \cdots \le u_{K-2} \le 1} \Big[\sum_{k=1}^{K-2} (S^k_{(u_k)} - S^{k+1}_{(u_k)} + p^k_{(u_k)(1)}) + \max_{u_{K-2} \le u_{K-1} \le N} \Big\{ \sum_{i=1}^{u_{K-1}-1} (Q^{K-1}_{(i)} - Q^K_{(i)}) + S^{K-1}_{(u_{K-1})} - S^K_{(u_{K-1})} + \max_{1 \le \nu_{K-1} \le n_{u_{K-1}}} \Big(\sum_{\xi=1}^{\nu_{K-1}} p^{K-1}_{(u_{K-1})(\xi)} - \sum_{\xi=1}^{\nu_{K-1}-1} p^K_{(u_{K-1})(\xi)} \Big) \Big\} \Big] \tag{4.41}$$

Setting $u_{K-2} = 1$ and $u_{K-2} = 0$, this is equivalent to the following:

$$I = \max \Big[0, \max_{0 \le u_1 \le u_2 \le \cdots \le u_{K-2} = 1} \sum_{k=1}^{K-2} (S^k_{(u_k)} - S^{k+1}_{(u_k)} + p^k_{(u_k)(1)}) + \max_{1 \le u_{K-1} \le N} \Big\{ \sum_{i=1}^{u_{K-1}-1} (Q^{K-1}_{(i)} - Q^K_{(i)}) + S^{K-1}_{(u_{K-1})} - S^K_{(u_{K-1})} + \max_{1 \le \nu_{K-1} \le n_{u_{K-1}}} \Big(\sum_{\xi=1}^{\nu_{K-1}} p^{K-1}_{(u_{K-1})(\xi)} - \sum_{\xi=1}^{\nu_{K-1}-1} p^K_{(u_{K-1})(\xi)} \Big) \Big\} \Big] \tag{4.42}$$

Hence, the objective function to be minimized becomes

$$I' = \max_{0 \le u_1 \le u_2 \le \cdots \le u_{K-2} = 1} \sum_{k=1}^{K-2} (S^k_{(u_k)} - S^{k+1}_{(u_k)} + p^k_{(u_k)(1)}) + \max_{1 \le \nu_{K-1} \le N} \Big\{ \sum_{i=1}^{u_{K-1}-1} (Q^{K-1}_{(i)} - Q^K_{(i)}) + S^{K-1}_{(u_{K-1})} - S^K_{(u_{K-1})} + \max_{1 \le \nu_{K-1} \le n_{u_{K-1}}} \Big(\sum_{\xi=1}^{\nu_{K-1}} p^{K-1}_{(u_{K-1})(\xi)} - \sum_{\xi=1}^{\nu_{K-1}-1} p^K_{(u_{K-1})(\xi)} \Big) \Big\} \tag{4.43}$$

In the above equation, the first term is concerned with the job occupying the first place in a group schedule, and the second one with the jobs being processed on the last two stages M_{K-1} and M_K. The group schedule, S, determined by applying Theorem 4.5 on the two-stage problem (stages M_{K-1} and M_K) minimizes the second term. Hence, the optimal group schedule can be obtained by evaluating the total elapsed times of the schedules, $S_{i\xi}$, generated by assigning, first, each group G_i containing the job $J_{i\xi}$ ($i=1, 2,\ldots, N$, $\xi=1, 2,\ldots, n_i$) to the first position of schedule S, and then placing each job $J_{i\xi}$ first in the job sequence of G_i, and maintaining the other group and job positions. Let $J_{a\alpha}$ be the first job of schedule S. Since schedule S minimizes the second term of equation (4.43), the optimal group schedule is determined by only examining the schedules $S_{i\xi}$ for which

$$\max_{0\le u_1\le u_2\le\cdots\le u_{K-2}=1} \sum_{k=1}^{K-2}\sum_{i=1}^{u_k}(S_i^k - S_i^{k+1} + p_{i\xi}^k) < \max_{0\le u_1\le u_2\le\cdots\le u_{K-2}=1} \sum_{k=1}^{K-2}\sum_{i=1}^{u_k}(S_a^k - S_a^{k+1} + p_{a\alpha}^k) \tag{4.44}$$

Thus, for Case 2, the optimizing algorithm for determining an optimal group schedule is proposed as follows:

< Optimizing algorithm for Case 2 >

Step 1. Determine an optimal group schedule, S, for the two-stage problem (stages M_{K-1} and M_K). Let $J_{a\alpha}$ be the first job of schedule S.

Step 2. Let $\Pi = (J_{i_1\xi_1}, J_{i_2\xi_2},\ldots, J_{i_l\xi_l})$ be a set of jobs such that

$$\max_{0\le u_1\le u_2\le\cdots\le u_{K-2}=1} \sum_{k=1}^{K-2}\sum_{i_j=1}^{u_k}(S_{i_j}^k - S_{i_j}^{k+1} + p_{i_j\xi_j}^k) < \max_{0\le u_1\le u_2\le\cdots\le u_{K-2}=1} \sum_{k=1}^{K-2}\sum_{i=1}^{u_k}(S_a^k - S_a^{k+1} + p_{a\alpha}^k)$$

Generate l new group schedules by assigning, first, each group containing the jobs $J_{i_j\xi_j}$ ($\in \Pi$) to the first position, and then placing each job $J_{i_j\xi_j}$ first in the job sequence of G_{i_j} and maintaining the schedule S order for the remaining groups and jobs.

Step 3. Among the $(l+1)$ schedules obtained above, find the group schedule minimizing the total elapsed time. This schedule is optimal.

For Case 3, the following relations hold.

$$H^k_{u\nu} \leq H^k_{u\nu+1} \quad \text{and} \quad H^k_{u-1n_{u-1}} \leq H^k_{u1} \quad (k=2,\ 3,\dots,\ K-1) \tag{4.45}$$

Thus

$$I = \max_{0\leq u_1\leq u_2=\dots=u_{K-1}=N} \ \max_{1\leq\nu_1\leq n_{u_1}} (H^1_{(u_1)(\nu_1)} + \sum_{k=2}^{K-1} H^k_{(N)(n_N)})$$

$$= \max_{0\leq u_1\leq N} \ \max_{1\leq\nu_1\leq n_{u_1}} \{ \sum_{i=1}^{u_1-1} (Q^1_{(i)} - Q^2_{(i)}) + S^1_{(u_1)} - S^2_{(u_1)}$$

$$+ \sum_{\xi=1}^{\nu_1} p^1_{(u_1)(\xi)} - \sum_{\xi=1}^{\nu_1-1} p^2_{(u_1)(\xi)} \} + \sum_{k=3}^{K} p^k_{(N)(n_N)} + \sum_{i=1}^{N} (Q^2_{(i)} - Q^K_{(i)}) \tag{4.46}$$

In the above equation, the first term is concerned with the jobs being processed on the first two stages M_1 and M_2, and the second one with the job occupying the last place in a group schedule. The third term is a constant.

For Case 3, the following algorithm is developed in much the same way as for Case 2.

< Optimizing algorithm for Case 3 >

Step 1. Determine an optimal group schedule, S, for the two-stage problem (stages M_1 and M_2). Let $J_{z\beta}$ be the last job of schedule S.

Step 2. Let $\Pi = (J_{i_1\xi_1},\ J_{i_2\xi_2},\dots,\ J_{i_l\xi_l})$ be a set of jobs such that

$$\sum_{k=3}^{K} p^k_{i_j\xi_j} < \sum_{k=3}^{K} p^k_{z\beta}$$

Generate l new group schedules by assigning, first, each group G_{i_j} containing the jobs $J_{i_j\xi_j}$ ($\in \Pi$) to the last position, and then placing each job $J_{i_j\xi_j}$ last in the job sequence of G_{i_j} and maintaining the schedule S order for the remaining groups and jobs.

Step 3. Among the $(l+1)$ schedules obtained above, find the group schedule minimizing the total elapsed time. This schedule is optimal.

4.4.2 Numerical examples

To illustrate the optimizing algorithms presented, consider two special structure flow-shop group scheduling problems.

Example 1: The production data for a 10-job, 4-group, 4-stage problem are given in Table 4.6. This is an example of Case 1, since the group setup times

Table 4.6 Production data for Example 1

(unit: hours)

Group	G_1			G_2				G_3			G_4			
Job		J_{11}	J_{12}		J_{21}	J_{22}	J_{23}		J_{31}	J_{32}		J_{41}	J_{42}	J_{43}
Setup time· processing time	s_1^k	p_{11}^k	p_{12}^k	s_2^k	p_{21}^k	p_{22}^k	p_{23}^k	s_3^k	p_{31}^k	p_{32}^k	s_4^k	p_{41}^k	p_{42}^k	p_{43}^k
Stage M_1	5	16	18	3	18	19	15	4	15	19	4	17	16	20
Stage M_2	4	15	13	3	12	14	11	3	10	15	5	10	13	15
Stage M_3	3	6	10	4	10	8	7	2	6	8	3	5	10	9
Stage M_4	4	13	20	4	15	6	17	5	18	20	4	12	9	17

and the job processing times satify conditions (4.27) for $h=3$ as follows:

$$\min_{1\le i\le 4}\,(s_i^1 - s_i^2 + \min_{1\le \xi\le n_i} p_{i\xi}^1)\,(=15) \ge \max_{1\le j\le 4}\ \max_{1\le \eta\le n_j} p_{j\eta}^2\,(=15)$$

$$\min_{1\le \xi\le 2} p_{1\xi}^1\ (=16) \ge \max_{1\le \eta\le 2} p_{1\eta}^2\ (=15)$$

$$\min_{1\le \xi\le 3} p_{2\xi}^1\ (=15) \ge \max_{1\le \eta\le 3} p_{2\eta}^2\ (=14)$$

$$\min_{1\le \xi\le 2} p_{3\xi}^1\ (=15) \ge \max_{1\le \eta\le 2} p_{3\eta}^2\ (=15)$$

$$\min_{1\le \xi\le 3} p_{4\xi}^1\ (=16) \ge \max_{1\le \eta\le 3} p_{4\eta}^2\ (=15)$$

$$\min_{1\le i\le 4}\,(s_i^2 - s_i^3 + \min_{1\le \xi\le n_i} p_{i\xi}^2)\,(=11) \ge \max_{1\le j\le N}\ \max_{1\le \eta\le n_j} p_{j\eta}^3\,(=10)$$

$$\min_{1\le \xi\le 2} p_{1\xi}^2\ (=13) \ge \max_{1\le \eta\le 2} p_{1\eta}^3\ (=10)$$

$$\min_{1 \le \xi \le 3} p^2_{2\xi} \ (=11) \ge \max_{1 \le \eta \le 3} p^3_{2\eta} \ (=10)$$

$$\min_{1 \le \xi \le 2} p^2_{3\xi} \ (=10) \ge \max_{1 \le \eta \le 2} p^3_{3\eta} \ (=8)$$

$$\min_{1 \le \xi \le 3} p^2_{4\xi} \ (=10) \ge \max_{1 \le \eta \le 3} p^3_{4\eta} \ (=10)$$

Thus the optimal group schedule is determined by using the optimizing algorithm for Case 1.

Step 1. The values of $x_{i\xi}$ and $y_{i\xi}$ for each job are computed as in Table 4.7. By applying Johnson's working rule to this table, the optimal job

Table 4.7 List of $X_{i\xi}$ and $y_{i\xi}$

Group	G_1		G_2			G_3		G_4		
Job	J_{11}	J_{12}	J_{21}	J_{22}	J_{23}	J_{31}	J_{32}	J_{41}	J_{42}	J_{43}
$x_{i\xi}$	37	41	40	41	33	31	42	32	39	43
$y_{i\xi}$	34	43	37	28	35	34	43	27	32	41

sequences for four groups are decided as J_{12}-J_{11}, J_{23}-J_{21}-J_{22}, J_{31}-J_{32}, and J_{43}-J_{42}-J_{41}, respectively.

Step 2. The values of X_i and Y_i for each group are calculated as $\{(X_1, Y_1), (X_2, Y_2), (X_3, Y_3), (X_4, Y_4)\} = \{(42, 40), (41, 28), (38, 39), (44, 29)\}$. For example, $X_1 = 5 - 4 + \max(41, 41 + 37 - 43) = 42$ and $X_2 = \max(43 + 34 - 37, 34) = 40$. Hence, from this list, the optimal group sequence is G_3-G_1-G_4-G_2.

Consequently, the optimal group schedule is determined as $G_3(J_{31}$-$J_{32})$-$G_1(J_{12}$-$J_{11})$-$G_4(J_{43}$-J_{42}-$J_{41})$-$G_2(J_{23}$-J_{21}-$J_{22})$ with the total elapsed time of 217 hours.

Example 2. The production data for a 10-job, 4-group, 4-stage problem are given in Table 4.8 The group setup times and the job processing times satisfy condition (4.28) of Case 2 as follows:

Table 4.8 Production data for Example 2

(units: hours)

Group	G_1			G_2				G_3			G_4			
Job		J_{11}	J_{12}		J_{21}	J_{22}	J_{23}		J_{31}	J_{32}		J_{41}	J_{42}	J_{43}
Setup time· processing time	s_1^k	p_{11}^k	p_{12}^k	s_2^k	p_{21}^k	p_{22}^k	p_{23}^k	s_3^k	p_{31}^k	p_{32}^k	s_4^k	p_{41}^k	p_{42}^k	p_{43}^k
Stage M_1	2	6	10	3	9	5	8	1	7	8	2	9	6	7
Stage M_2	3	14	12	2	10	11	15	2	13	11	2	13	5	10
Stage M_3	4	18	16	3	17	19	18	2	16	15	4	18	20	19
Stage M_4	2	9	19	4	22	12	25	3	20	23	3	9	13	10

$$\max_{1\le i\le 4}(s_i^1 - s_i^2 + \max_{1\le\xi\le n_i} p_{i\xi}^1)(=10) \le \min_{1\le j\le 4}\ \min_{1\le\eta\le n_j} p_{j\eta}^2(=10)$$

$$\max_{1\le\xi\le 2} p_{1\xi}^1(=10) \le \min_{1\le\eta\le 2} p_{1\eta}^2(=12)$$

$$\max_{1\le\xi\le 3} p_{2\xi}^1\ (=9) \le \min_{1\le\eta\le 3} p_{2\eta}^2(=10)$$

$$\max_{1\le\xi\le 2} p_{3\xi}^1\ (=8) \le \min_{1\le\eta\le 2} p_{3\eta}^2(=11)$$

$$\max_{1\le\xi\le 3} p_{4\xi}^1\ (=9) \le \min_{1\le\eta\le 3} p_{4\eta}^2(=10)$$

$$\max_{1\le i\le 4}(s_i^2 - s_i^3 + \max_{1\le\xi\le n_i} p_{i\xi}^2)(=14) \le \min_{1\le j\le 4}\ \min_{1\le\eta\le n_j} p_{j\eta}^3(=15)$$

$$\max_{1\le\xi\le 2} p_{1\xi}^2(=14) \le \min_{1\le\eta\le 2} p_{1\eta}^3(=16)$$

$$\max_{1\le\xi\le 3} p_{2\xi}^2(=15) \le \min_{1\le\eta\le 3} p_{2\eta}^3(=17)$$

$$\max_{1\le\xi\le 2} p_{3\xi}^2(=13) \le \min_{1\le\eta\le 2} p_{3\eta}^2(=15)$$

$$\max_{1\le\xi\le 3} p_{4\xi}^2(=15) \le \min_{1\le\eta\le 3} p_{4\eta}^3(=18)$$

Hence, the optimal group schedule is obtained by using the optimizing algorithm for Case 2.

Step 1. The optimal group schedule for the two-stage problem (stages M_3

and M_4) is determined as $G_3(J_{32}-J_{31})-G_2(J_{21}-J_{23}-J_{22})-G_1(J_{12}-J_{11})-G_4(J_{42}-J_{43}-J_{41})$. Thus $J_{a\alpha}=J_{32}$.

Step 2. The values of $\max_{0\le u_1\le u_2=1}\sum_{k=1}^{2}\sum_{i=1}^{u_k}(S_i^k - S_{i\xi}^{k+1}+p_{i\xi}^k)(=z_{i\xi})$ for each job are computed as $\{(z_{11}, z_{12}), (z_{21}, z_{22}, z_{23}), (z_{31}, z_{32}), (z_{41}, z_{42}, z_{43})\}=\{(18, 20), (19, 16, 23), (19, 18), (20, 19, 15)\}$. For example, $z_{11}=\max(S_1^2-S_1^3+p_{11}^2,\ S_1^1-S_1^2+p_{11}^1+S_1^2-S_1^3+p_{11}^2)=18$. $\Pi=\{J_{22}, J_{43}\}$, since $z_{32}=18$. Thus it is necessary to evaluate the following three schedules to determine the optimal group schedule.

Schedule S: $G_3(J_{32}-J_{31})-G_2(J_{21}-J_{23}-J_{22})-G_1(J_{12}-J_{11})-G_4(J_{42}-J_{43}-J_{41})$

Schedule S_{22}: $G_2(J_{22}-J_{21}-J_{23})-G_3(J_{32}-J_{31})-G_1(J_{12}-J_{11})-G_4(J_{42}-J_{43}-J_{41})$

Schedule S_{43}: $G_4(J_{43}-J_{42}-J_{41})-G_3(J_{32}-J_{31})-G_2(J_{21}-J_{23}-J_{22})-G_1(J_{12}-J_{11})$

Step 3. Since the total elapsed times for the above three schedules are 216, 212, and 229 hours, respectively, the optimal group schedule is the schedule S_{22}.

4.5 Conclusions

(1) The two-stage flow-shop scheduling model with setup times separated and time lags was developed, and the well-known Johnson's theorem for the two-stage problem of minimizing the total elapsed time was extended to the presented model.

(2) The two-stage flow-shop group scheduling problem was treated under the minimum total-elapsed-time criterion, and a theorem was given for determining the optimal group schedule.

(3) The multistage flow-shop group scheduling problem was considered under the same criterion, and a theoretical analysis was made into the special cases where there exist some well-defined relationships between the group setup times and the job processing times. For each case, a theorem or an algorithm was given to determine the optimal group schedule.

CHAPTER 5 GROUP SCHEDULING ON MULTIPLE PRODUCTION STAGES — BRANCH-AND-BOUND APPROACH

5.1 Introduction

As stated in the previous chapter, a successful analysis of multistage scheduling problems is limited to the case of the two-stage flow-shop problem with the objective of minimizing the total elapsed time. For more than three-stage scheduling problems, a universal theoretical analysis cannot be made even under the simple criterion of the minimum total elapsed time. In order to generally solve the problems, therefore, it is necessary to resort to general purpose methodologies, such as a dynamic programming approach, a branch-and-bound method, etc., or a heuristic procedure.

Among these, the branch-and-bound method has been employed with some success.[1) The basic branch-and-bound procedure for solving the three-stage flow-shop scheduling problem of minimizing the total elapsed time was developed by Ignall and Schrage,[2) and independently by Lomnicki.[3) Since then, a variety of extensions and refinements have been developed for the branch-and-bound procedure.[4,5)

An attempt to solve the flow-shop group scheduling problem using the branch-and-bound method was made by Nakamura and Hitomi.[6) The main purpose of this study is to find an optimal group schedule and no attention is directed to the effectiveness of the lower bounds proposed and the optimizing algorithm developed.

With the help of the branch-and-bound method, this chapter also solves the flow-shop group scheduling problems.[7) In these problems, the ordering of groups and jobs is assumed to be the same on each machine. The scheduling criteria employed are the minimization of the total elapsed time and the minimization of the weighted mean flow time for which no effective lower bound has been developed for the multistage scheduling problem. In this chapter, attention is paid to the effectivenesses of the lower bounds to be proposed and the optimizing algorithms to be developed. For the minimum-total-elapsed-time problem, three kinds of lower bounds are developed by extending the typical ones in the conventional scheduling to the group scheduling. Then an optimizing algorithm which incorporates these bounds is proposed. Their relative effectiveness is then investigated with numerical experiments.

For the minimum-weighted-mean-flow-time problem, a lower bound, which is an extension of the machine-based bound, is developed, and numerical experiments are run to examine the effect of randomness of weighting factors given to jobs on the effectiveness of the algorithm.

5.2 Total elapsed time and weighted mean flow time

As is defined in the previous chapters, let $p^k_{i\xi}$ ($i = 1, 2,\ldots, N$, $\xi = 1, 2,\ldots, n_i$, $k = 1, 2,\ldots, K$) and s^k_i ($i = 1, 2,\ldots, N$, $k = 1, 2,\ldots, K$) denote the job processing time including the job setup time of job $J_{i\xi}$ ($i = 1, 2,\ldots, N$, $\xi = 1, 2,\ldots, n_i$) of group G_i ($i=1, 2,\ldots, N$) on stage (machine) M_k ($k = 1, 2,\ldots, K$) and the group setup time of group G_i on stage M_k, respectively. Furthermore, let $w_{i\xi}$ ($i=1, 2,\ldots, N$, $\xi=1, 2, \ldots, n_i$) be a weighting factor given to job $J_{i\xi}$.

The total elapsed time required to complete all jobs in all groups is denoted as in equation (4.26) in Chapter 4. Another expression of the total elapsed time can be given by using the idle times of jobs on the last stage.

The completion time on M_k of $J_{(i)(\xi)}$ indicating the ξth job in the ith group in a group schedule is denoted as

$$C^k_{(i)(\xi)} = \sum_{j=1}^{i-1} (\sum_{\eta=1}^{n_j} g^k_{(j)(\xi)} + S^k_{(j)} + P^k_{(j)}) + S^k_{(i)}$$

$$+ \sum_{\eta=1}^{\xi} (g^k_{(i)(\eta)} + p^k_{(i)(\eta)}) \tag{5.1}$$

where $P^k_{(i)} = \sum_{\xi=1}^{n_i} p^k_{(i)(\xi)}$ and $g^k_{(i)(\xi)}$ is the idle time of M_k before processing ξth job after completion of $(\xi-1)$th job of $G_{(i)}$, and is given as follows:

for $\xi \neq 1$,

$$g^k_{(i)(\xi)} = \begin{cases} C^{k-1}_{(i)(\xi)} - C^k_{(i)(\xi-1)}, & \text{if } C^k_{(i)(\xi)} > C^k_{(i)(\xi-1)} \\ 0, & \text{otherwise} \end{cases}$$

for $\xi = 1$,

$$g^k_{(i)(1)} = \begin{cases} C^{k-1}_{(i)(1)} - C^k_{(i-1)(n_{i-1})} - S^k_{(i)}, & \text{if } C^{k-1}_{(i)(1)} > C^k_{(i-1)(n_{i-1})} + S^k_{(i)} \\ 0, & \text{otherwise} \end{cases}$$

Hence, the total elapsed time is given by

$$\begin{aligned} F_{max} &= C^K_{(N)(n_N)} \\ &= \sum_{i=1}^{N} (\sum_{\xi=1}^{n_i} g^K_{(i)(\xi)} + S^K_{(i)} + P^K_{(i)}) \end{aligned} \tag{5.2}$$

The weighted mean flow time is given by

$$\bar{F}_w = \sum_{i=1}^{N} \sum_{\xi=1}^{n_i} w_{(i)(\xi)} C^K_{(i)(\xi)} \Big/ \sum_{i=1}^{N} n_i$$

$$= \{ \sum_{i=1}^{N} \sum_{\xi=1}^{n_i} w_{(i)(\xi)} (\sum_{\xi=1}^{n_i} g^K_{(i)(\xi)} + S^K_{(i)} + P^K_{(i)}) \} / M \tag{5.3}$$

where $M = \sum_{i=1}^{N} n_i$.

Equation (5.3) indicates the mean flow time in the case in which all the weighting factors are equal to one.

5.3 Branch-and-bound method for multistage group scheduling

The branch-and-bound method, which is one of the optimizing techniques, has been employed to solve flow-shop scheduling problems and has shown some success. In order to solve a flow-shop group scheduling problem, this method can be applied. The branching procedure, which is one of the fundamental procedures of the branch-and-bound method, is done in the same way as in the case of the single-stage group scheduling mentioned in Chapter 3. In this section, therefore, the bounding procedure, which is another fundamental procedure, is explained.

The calculation of the lower bound is made at each of the job nodes created by the branching procedure. Let N_{rs} be a job node at which s jobs selected among $n_{(r)}$ jobs in group $G_{(r)}$ at a group node N_r are allocated. Two kinds of lower bounds are estimated according to the criterion employed.

(1) Lower bounds on the total elapsed time

A variety of lower bounds on the total elapsed time have been developed for the conventional flow-shop scheduling problem. Representative of these bounds are the machine-based, the job-based, and the composite bounds.[5] For determining the optimal group schedule, the lower bounds can be developed by extending the above three bounds to the group scheduling, as follows:

(a) Machine-based bound: The machine-based bound at N_{rs} is estimated by

$$L_1(N_{rs}) = \max_{1\leq k\leq K} \{C^k_{(r)(s)} + \sum_{\xi\in\bar{J}_r} p^k_{(r)(\xi)} + \sum_{i\in\bar{G}_r} (S^k_{(i)} + P^k_{(i)}) + \min_{i\xi\in\bar{J}_{rs}} \sum_{h=k+1}^{K} p^h_{(i)(\xi)}\} \tag{5.4}$$

where $C^k_{(r)(s)}$ is the completion time of $J_{(r)(s)}$ on M_k, and $\bar{G}_r$ is the set of groups not yet sequenced, and $\bar{J}_{rs}$ and $\bar{J}_r$ are the set of jobs not yet sequenced and the set of jobs not yet sequenced in group $G_{(r)}$, respectively.

The second and third terms of the above equation are the sum of the job processing times for jobs not yet sequenced in the current group $G_{(r)}$ and the sum of the group processing times for groups not yet specified in the node, respectively. The last one represents the minimum of the sums of job processing times in the remaining stages for each of the jobs not yet sequenced.

(b) Job-based bound: The job-based bound at N_{rs} is given by

$$L_2(N_{rs}) = \max_{1 \le k \le K} [C^k_{(r)(s)} + \max_{i\xi \in \bar{J}_{rs}} \{ \sum_{h=k}^{K} p^h_{(i)(\xi)} + \sum_{j\eta \in \bar{J}_{rs}} \min_{\substack{j \ne i \\ \eta \ne \xi}} (p^k_{(j)(\eta)}, p^K_{(j)(\eta)}) + \sum_{j \in \bar{G}_r} \min(S^k_{(j)}, S^K_{(j)}) \}] \quad (5.5)$$

This bound expresses the fact that the total elapsed time may be determined by the total processing time for a job rather than the total processing time on a machine.

(c) Composite lower bound: The composite lower bound, which is a combination of the above two bounds, is

$$L_3(N_{rs}) = \max \{ L_1(N_{rs}), L_2'(N_{rs}) \} \quad (5.6)$$

where $L_2'(N_{rs})$ is obtained by eliminating the bound on M_K in equation (5.5).

(2) Lower bound on the weighted mean flow time

Even for the conventional multistage scheduling problem, very few reports have been made on the lower bound on the weighted mean flow time.[5] The lower bound for the group scheduling problem is estimated by extending the equation (5.4) to the case of the weighted mean flow time.

The lower bound at N_{rs} is calculated by

$$L_w(N_{rs}) = \{ W_1(N_{rs}) + W_2(N_{rs}) \} / \sum_{i=1}^{N} n_i \quad (5.7)$$

where $W_1(N_{rs})$ and $W_2(N_{rs})$ are the weighted flow times for groups and jobs which are already sequenced, and not sequenced, respectively, and are given by

$$W_1(N_{rs}) = \sum_{i=1}^{r-1} \sum_{\xi=1}^{n_i} w_{(i)(\xi)} C^K_{(i)(\xi)} + \sum_{\xi=1}^{s} w_{(r)(\xi)} C^K_{(r)(\xi)} \quad (5.8)$$

$$W_2(N_{rs}) = \max_{1\le k\le K} [\sum_{\xi=s+1}^{n_r} w_{(r)(\xi)} (C^k_{(r)(s)} + \sum_{\eta=s+1}^{\xi} p^k_{(r)(\eta)} + \sum_{h=k+1}^{K} p^h_{(r)(\xi)})$$

$$+ \sum_{i=r+1}^{N} \sum_{\xi=1}^{n_i} w_{(i)(\xi)} \{ C^k_{(r)(n_r)} + \sum_{j=r+1}^{i-1} (S^k_{(j)} + P^k_{(j)}) + S^k_{(i)}$$

$$+ \sum_{\eta=1}^{\xi} p^k_{(i)(\eta)} + \sum_{h=k+1}^{K} p^h_{(i)(\xi)} \}] \qquad (5.9)$$

Equation (5.8) is obvious. In equation (5.9), the first term in the 'max' operation is for jobs not yet sequenced in the current group $G_{(r)}$, and is introduced in the following way:

The lower bound of the completion time $C'_{(r)(\xi)}$ of $J_{(r)(\xi)}$ $(\xi > s)$ is given as follows by using the completion time $C^k_{(r)(s)}$ on M_k of $J_{(r)(s)}$, which is the last job in the sequence already sequenced.

$$C'_{(r)(\xi)} = C^k_{(r)(s)} + \sum_{\eta=s+1}^{\xi} p^k_{(r)(\eta)} + \sum_{h=k+1}^{K} p^h_{(r)(\xi)} \qquad (5.10)$$

Since the total weighted flow time for $(n_{(r)} - s)$ jobs not yet sequenced is $\sum_{\xi=s+1}^{n_r} w_{(r)(\xi)} C'_{(r)(\xi)}$, the first term is obtained. The second term in the 'max' operation for the jobs in $(N - r)$ groups not yet sequenced is obtained in much the same way as the first term.

The value of equation (5. 9) depends on the sequences of groups and jobs not yet sequenced. For determining the minimum value of equation (5. 9) Theorem 2.2 offered in Chapter 2 is useful. This theorem gives a group schedule minimizing the weighted mean flow time for the single-stage group scheduling problem. In order to give the minimum value to equation (5. 9), the equation is calculated so that groups and jobs not yet sequenced are supposed to be ordered in each stage according to the order determined by Theorem 2.2.

Based on the analytical results above, the optimizing branch-and-bound algorithm for determining the optimal group schedule under the criterion of the minimum total elapsed time or the minimum weighted mean flow time is proposed as follows:

< Optimizing algorithm based on the branch-and-bound method >

Step 1. Let the group level $r=0$ and the least feasible value $L^*=\infty$. Go to Step 2.

Step 2. Branch the group node N_r into $(N-r)$ group nodes N_{r+1} by placing each of the groups not yet allocated next in the sequence determined. Set $r=r+1$, then go to Step 3.

Step 3. For each of the group nodes N_r, create job nodes N_{rs} of the job level $s=1$ by placing each of the jobs in the group next in the sequence determined. Go to Step 4.

Step 4. Calculate the lower bound $LB(N_{rs})$ for each of the new job nodes N_{rs} by using equation (5.4), (5.5), (5.6), or (5.7) depending on the scheduling criterion employed.

Step 5. Find the job node having min $LB(N_{rs})$ from among the job nodes derived in Step 3 or 8 in the case of $L^* = \infty$, or from among all job nodes being active in the case of $L^* \neq \infty$. (In the case of a tie, select the node with the largest value of, first, r, and then s.) Let the group level and job level of the node be r and s, respectively, and $LB^*(N_{rs}) = LB(N_{rs})$. Go to Step 6.

Step 6. If $LB^*(N_{rs}) < L^*$, then go to Step 7. Otherwise, stop. (The group and job sequences of the node having L^* are optimal.)

Step 7. If $s < n_{(r)}$, then go to Step 8. Otherwise go to Step 9.

Step 8. Branch the job node N_{rs} into $(n_{(r)}-s)$ nodes N_{rs+1} by placing each of the jobs not yet allocated in group $G_{(r)}$ next in the sequence determined. Set $s=s+1$, then go back to Step 4.

Step 9. If $r < N$, then go back to Step 2. Otherwise, $L^* = LB^*(N_{rs})$, so go back to Step 5.

5.4 Numerical experiments

5.4.1 Numerical example

In an attempt to clarify the features of the multistage group scheduling problem, a simple numerical example will be given. The basic data for a 10-job, 3-group, 4-stage problem are given in Table 5.1.

Table 5.1 Production data for group scheduling

(units: min)

Group	G_1				G_2					G_3			
Job		J_{11}	J_{12}	J_{13}		J_{21}	J_{22}	J_{23}	J_{24}		J_{31}	J_{32}	J_{33}
Setup time· processing time	s_1^k	p_{11}^k	p_{12}^k	p_{13}^k	s_2^k	p_{21}^k	p_{22}^k	p_{23}^k	p_{24}^k	s_3^k	p_{31}^k	p_{32}^k	p_{33}^k
Stage M_1	10	35	36	51	25	41	16	31	32	29	35	41	17
Stage M_2	26	36	36	49	17	28	34	13	34	12	47	19	30
Stage M_3	12	46	34	22	26	49	13	29	50	18	35	37	46
Stage M_4	30	48	27	41	14	22	20	49	39	15	38	24	33

The optimal group schedule which minimizes the total elapsed time is determined by the optimizing algorithm proposed. Fig. 5.1 shows the branching tree in the case where the machine-based bound is used in the algorithm. The lower bound of the total elapsed time for each job node is given just below the corresponding node in the figure. The optimal group schedule is $G_2(J_{23}-J_{24}-J_{22}-J_{21})-G_1(J_{11}-J_{12}-J_{13})-G_3(J_{33}-J_{31}-J_{32})$ with the total elapsed time of 518 min.

5.4.2 Numerical experiments

In order to examine the effectiveness of the optimizing algorithm proposed, the algorithm was programmed in FORTRAN and a TOSBAC 5600 computer was used. Numerical experiments were run for 20 group-and-job sets which

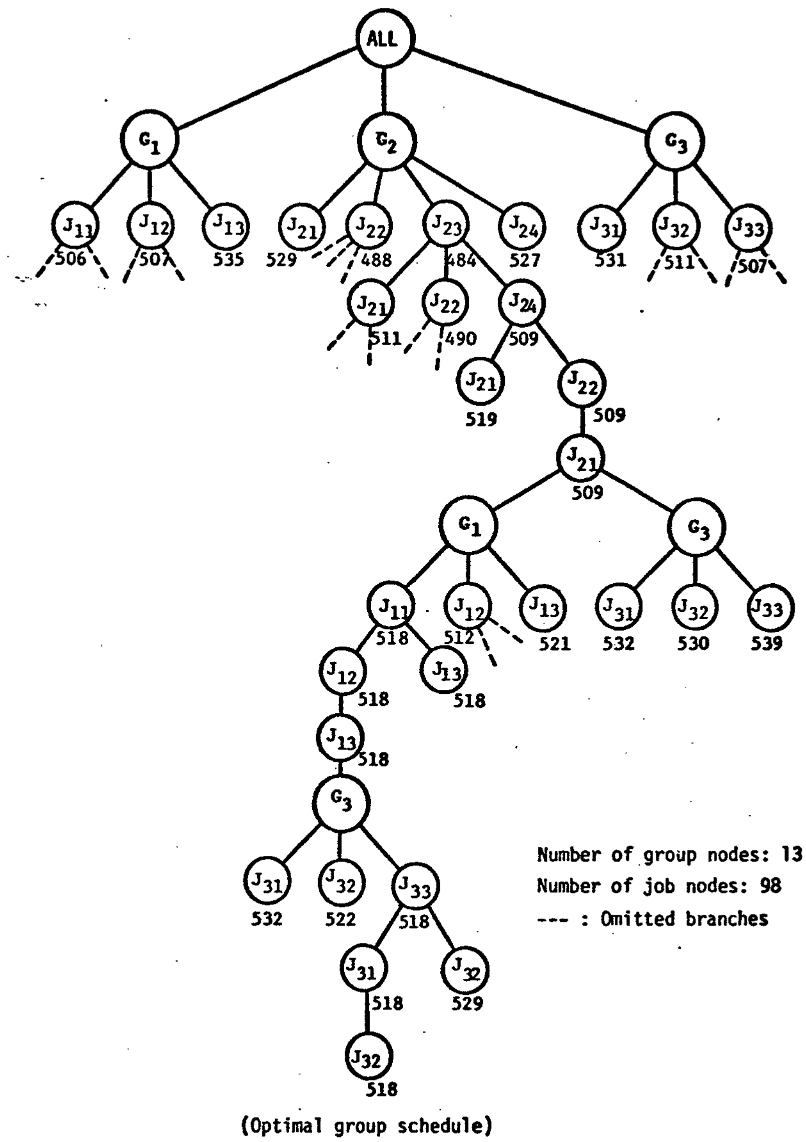


Fig. 5.1 The branching tree for the example problem

consisted of two to four groups with two to four jobs in each group. The number of stages is set at four. The job processing times and the group setup times are obtained from uniform distributions ranging from 10 to 55 and 10 to 30, respectively.

(1) Results for the minimum total elapsed time

The computational results for the three kinds of lower bounds are shown in Table 5.2. Fig. 5.2 shows the relative effectiveness of the three lower bounds for various sizes of problems. It is well known that the composite lower bound is more efficient as compared with the

Table 5.2 Computational results for the minimum-total-elapsed-time criterion

Problem No.	1			2			3*	
Problem Size**	(3)(4)			(3)(4)(3)			(2)(3)(3)(4)	
Lower bound	Machine -based	Job-based	Composite	Machine -based	Job-based	Composite	Machine -based	Composite
Average CPU time (sec)	2.9	2.6	2.0	10.4	32.8	11.4	21.2	50.0
Average number of nodes	81	82	64	259	370	219	349	310
Maximum number of nodes	132	199	114	719	870	598	1105	933
Minimum number of nodes	31	31	31	62	62	62	45	45

* One problem is deleted, since more than 5000 nodes were created.

** For example, size (2)(3)(3)(4) shows that the number of groups is 4 and each group consists of 2, 3, 3, and 4 jobs, respectively.

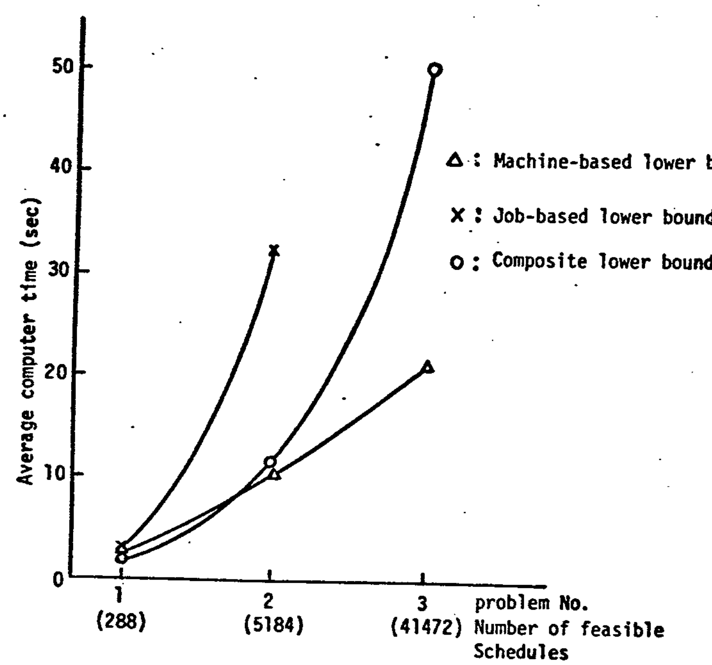


Fig. 5.2 Average computer times for three kinds of lower bounds

machine-based and the job-based bounds for the conventional multistage scheduling. In the group scheduling, however, Fig. 5.2 shows that the machine-based bound is more effective than the others as the sizes of the problems become large.

(2) Results for the minimum weighted mean flow time

In this experiment, two cases concerning the weighting factors are examined. One is the case in which equal weighting factors are assigned to all jobs, and the other is the case in which random numbers ranging from 1 to 5 are given to the jobs as the weighting factors. Table 5.3 shows the computational results for both cases. Fig. 5.3 indicates the variation

Table 5.3 Computational results for the minimum-weighted-mean-flow-time criterion

Problem No.	1		2		3		4	
Problem Size	(3)(4)		(3)(4)(3)		(2)(3)(3)(4)		(3)(4)(3)(4)	
Lower bound	$\bar{F}$	$\bar{F}_w$	$\bar{F}$	$\bar{F}_w$	$\bar{F}$	$\bar{F}_w$	$\bar{F}$	$\bar{F}_w$
Average CPU time (sec)	0.9	0.7	7.5	2.8	22.0	13.8	94.6	32.7
Average number of nodes	43	36	165	90	394	238	1561	539
Maximum number of nodes	64	62	268	189	746	552	4811	1661
Minimum number of nodes	26	20	73	50	108	80	657	114

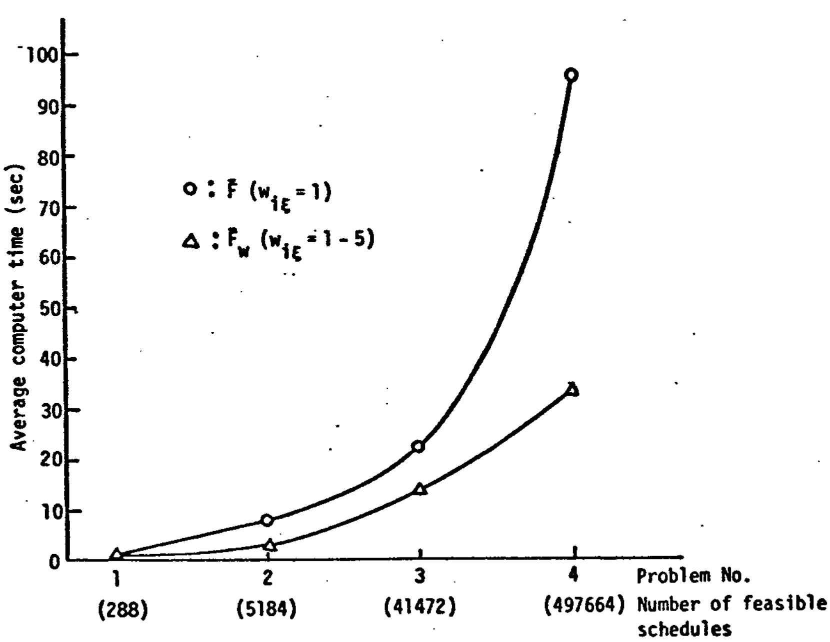


Fig. 5.3 Average computer times for lower bounds on the weighted mean flow time

of CPU time required to determine the optimal group schedule for different sizes of problems.

In the case of equal weighting factors, CPU time increases sharply as the sizes of problems become large. On the other hand, the increase of CPU time in the case of random weighting is relatively slow as compared with that in the case of equal weighting. This can be explained by the fact that the differences in the weighted flow times for jobs may be large due to the variation of the weighting factors.

5.5 Conclusions

(1) The branch-and-bound method was applied for solving the multistage group scheduling problems under the minimum-total-elapsed-time and the minimum-weighted-mean-flow-time criteria.

(2) The optimizing algorithm for determining the optimal group schedule was developed.

(3) The effectiveness of the optimizing algorithm proposed was tested with a numerical example.

(4) The machine-based bound was found more effective as compared with the job-based and the composite lower bounds for the group scheduling under the minimum-total-elapsed-time criterion. For the group scheduling under the minimum-weighted-mean-flow-time criterion, the CPU time required to determine the optimal group schedule was small when the variation of the weighting factors was large.

CHAPTER 6 OPTIMAL GROUP SCHEDULING AND MACHINING-SPEED DECISION UNDER DUE-DATE CONSTRAINTS

6.1 Introduction

In the field of production scheduling, the processing time required to complete a specified operation of a job is set at a constant in most cases. In practical situations, however, it is possible to vary the processing times by actively changing manufacturing conditions, especially machining speeds. In these cases, some modifications must be made on the scheduling model.

Based on such a consideration, Hitomi[1] has developed a production scheduling model with variable processing times depending on machining conditions. This model has been extended to the group scheduling model.[2,3] In this model, an optimal group schedule minimizing the total elapsed time was determined and then optimal machining speeds were decided so as to minimize the total production cost under the minimum-total-elapsed-time schedule.

In production scheduling problems, there are many kinds of criteria by which schedules are evaluated.[4,5] Under the actual situation of determining the processing order of jobs, meeting the jobs' due dates is one of the most important factors. In this chapter, under due-date constraints, a group scheduling model is developed on the multiple production stages with variable processing times and production costs.[6] Among the scheduling criteria concerning due dates, the criterion of minimizing the number of tardy jobs is employed in the model. In general, there exist a lot of schedules with the minimum number of tardy jobs. Therefore, it is necessary to select an optimal schedule by another criterion. The criterion employed as a secondary one is minimization of the total elapsed time.[7] Once a group schedule minimizing the total elapsed time with the minimum number of tardy jobs is determined, the optimal machining conditions which minimize the total production cost are determined by utilizing the idle times of the schedule determined.

6.2 Group scheduling model with variable processing times and costs

6.2.1 Assumptions and optimizing criteria

In addition to the fundamental assumptions of the group scheduling model defined in Chapter 1, the following ones are made in an attempt to construct a model with variable processing times and costs.

(i) Job processing time consists of job setup time and unit production time multiplied by lot size.

(ii) Unit production time and cost are dependent on machining condition (machining speed).

In this model, two kinds of scheduling criteria are adopted for group scheduling. The primary criterion is minimization of the number of tardy jobs. The secondary one is minimization of the total elapsed time. The criterion of determining machining speeds is the minimization of the total production cost which is a function of the machining speed.

6.2.2 Job processing time and production cost

Let $O_{i\xi}^{k}$ ($i = 1, 2, \ldots, N$, $\xi = 1, 2, \ldots, n_i$, $k = 1, 2, \ldots, K$) be kth operation on stage M_k ($k = 1, 2, \ldots, K$) of job $J_{i\xi}$ ($i = 1, 2, \ldots, N$, $\xi = 1, 2, \ldots, n_i$) of group G_i ($i = 1, 2, \ldots, N$). Unit production time, $u_{i\xi}^{k}$ (min/pc), of $O_{i\xi}^{k}$ is expressed as a function of machining speed, $v_{i\xi}^{k}$ (m/min), for this operation as follows:[8]

$$u_{i\xi}^{k} = a_{i\xi}^{k} + t_{i\xi}^{k} + b_{i\xi}^{k}\frac{t_{i\xi}^{k}}{T_{i\xi}^{k}}$$

$$= a^k_{i\xi} + \frac{\lambda^k_{i\xi}}{v^k_{i\xi}} + \frac{\lambda^k_{i\xi} b^k_{i\xi}}{(C^k_{i\xi})^{1/n^k_{i\xi}}} (v^k_{i\xi})^{1/n^k_{i\xi} - 1} \tag{6.1}$$

$$(i = 1, 2, \ldots, N,\ \xi = 1, 2, \ldots, n_i,\ k = 1, 2, \ldots, K)$$

where $a^k_{i\xi}$ is the preparation time (min/pc), $b^k_{i\xi}$ is the tool replacement time (min/edge), $t^k_{i\xi}$ is the actual machining time (min/pc), $T^k_{i\xi}$ is the tool life (min/edge), $\lambda^k_{i\xi}$ is the machining constant, $n^k_{i\xi}$ and $C^k_{i\xi}$ are the parameters for the Taylor tool-life equation for $O^k_{i\xi}$.

When $J_{i\xi}$ is processed in a lot size $l_{i\xi}$, the job processing time of $J_{i\xi}$ on M_k is given by

$$p^k_{i\xi} = s^k_{i\xi} + l_{i\xi} u^k_{i\xi} \tag{6.2}$$

where $s^k_{i\xi}$ is the job setup time of $J_{i\xi}$ on M_k.

Then the group processing time of G_i on M_k is

$$Q^k_i = S^k_i + P^k_i \tag{6.3}$$

where S^k_i is the group setup time of G_i on M_k and $P^k_i = \sum_{\xi=1}^{n_i} p^k_{i\xi}$.

Unit production cost, $q^k_{i\xi}$ (\$/pc), of $O^k_{i\xi}$ is expressed as a function of the machining speed $v^k_{i\xi}$ (m/min) as follows:[8]

$$q^k_{i\xi} = \alpha^k_{i\xi} a^k_{i\xi} + (\alpha^k_{i\xi} + \beta^k_{i\xi}) t^k_{i\xi} + (\alpha^k_{i\xi} b^k_{i\xi} + \varepsilon^k_{i\xi}) \frac{t^k_{i\xi}}{T^k_{i\xi}}$$

$$= \alpha^k_{i\xi} a^k_{i\xi} + (\alpha^k_{i\xi} + \beta^k_{i\xi}) \frac{\lambda^k_{i\xi}}{v^k_{i\xi}}$$

$$+ (\alpha^k_{i\xi} b^k_{i\xi} + \varepsilon^k_{i\xi}) \frac{\lambda^k_{i\xi}}{(C^k_{i\xi})^{1/n^k_{i\xi}}} (v^k_{i\xi})^{1/n^k_{i\xi} - 1} \tag{6.4}$$

$$(i = 1, 2, \ldots, N,\ \xi = 1, 2, \ldots, n_i,\ k = 1, 2, \ldots, K)$$

where $\alpha^k_{i\xi}$ is the direct labor cost and overhead (\$/min), $\beta^k_{i\xi}$ is the machining overhead (\$/min), and $\varepsilon^k_{i\xi}$ is the tool cost (\$/edge) for $O^k_{i\xi}$.

Introduce the maximum-production-rate machining speed and the minimum-production-cost machining speed, which will play important roles in determining the optimal group schedule and the optimal machining speeds. The maximum-production-rate machining speed for $O^k_{i\xi}$ is determined by setting the derivative of equation (6.1) in regard to $v^k_{i\xi}$ equal to zero:

$$v^{k(t)}_{i\xi} = C^k_{i\xi} \,/\, \{(\frac{1}{n^k_{i\xi}} - 1)b^k_{i\xi}\}^{n^k_{i\xi}} \tag{6.5}$$

$$(i=1,\ 2,\ldots,\ N,\ \xi=1,\ 2,\ldots,\ n_i,\ k=1,\ 2,\ldots,\ K)$$

With this machining speed the minimum job processing time of $J_{i\xi}$ on M_k is obtained by

$$p^{k(t)}_{i\xi} = s^k_{i\xi} + l^k_{i\xi}[a^k_{i\xi} + \frac{\lambda^k_{i\xi}}{(1-n^k_{i\xi})C^k_{i\xi}}\{(\frac{1}{n^k_{i\xi}} - 1)b^k_{i\xi}\}^{n^k_{i\xi}}] \tag{6.6}$$

$$(i=1,\ 2,\ldots,\ N,\ \xi=1,\ 2,\ldots,\ n_i,\ k=1,\ 2,\ldots,\ K)$$

The minimum-production-cost machining speed is obtained by setting the derivative of equation (6.4) in regard to $v^k_{i\xi}$ equal to zero:

$$v^{k(c)}_{i\xi} = C^k_{i\xi}\{(\frac{n^k_{i\xi}}{1 - n^k_{i\xi}})(\frac{\alpha^k_{i\xi} + \beta^k_{i\xi}}{\alpha^k_{i\xi}b^k_{i\xi} + \varepsilon^k_{i\xi}})\}^{n^k_{i\xi}} \tag{6.7}$$

$$(i=1,\ 2,\ldots,\ N,\ \xi=1,\ 2,\ldots,\ n_i,\ k=1,\ 2,\ \ldots,\ K)$$

6.2.3 Total elapsed time, number of tardy jobs, and total production cost

The total elapsed time is given by equation (5.2) in Chapter 5 as follows:

$$F_{max} = \sum_{i=1}^{N}(\sum_{\xi=1}^{n_i} g^K_{(i)(\xi)} + S^K_{(i)} + P^K_{(i)}) \tag{5.3}$$

where $g^K_{(i)(\xi)}$ is the idle time of M_K before processing ξth job after completion of $(\xi-1)$th job of $G_{(i)}$.

The number of tardy jobs is given by

$$N_T = \sum_{i=1}^{N} \sum_{\xi=1}^{n_i} \delta(T_{(i)(\xi)}) \tag{6.8}$$

where $T_{(i)(\xi)}$ is the tardiness of $J_{(i)(\xi)}$ and

$$\delta(x) = \begin{cases} 1, & x > 0 \\ 0, & \text{otherwise} \end{cases}$$

In addition to the unit production cost given by equation (6.4), the cost required for group setup is involved in the total production cost. However, the group setup cost is independent of the machining speed which is a decision variable. Therefore, it may be excluded in the total production cost which is a performance measure in determining the optimal machining speeds.

The total production cost to be minimized is given by

$$C_T = \sum_{i=1}^{N} \sum_{\xi=1}^{n_i} \sum_{k=1}^{K} l^k_{i\xi} q^k_{i\xi} \tag{6.9}$$

6.3 Determining optimal group schedule

6.3.1 Initial machining speeds and job processing times

In order to minimize the total elapsed time with the minimum number of tardy jobs, it is reasonable to set machining speeds at the maximum production rate initially, since both performance measures are nondecreasing functions of the completion time of each job. In determining the optimal group schedule, therefore, the time given by equation (6.6) is used as the processing time for each job.

6.3.2 Branch-and-bound method for group scheduling

A branch-and-bound method is applied to determine the schedule minimizing the total elapsed time with the minimum number of tardy jobs in the same way as in the previous chapters. Two fundamental procedures for solving the current problem are as follows:

(1) Branching procedure

The branching procedure for the current group scheduling problem with the dual scheduling criteria is fundamentally the same as the one for the problem with a single criterion in Chapter 5. However, there exists a difference between the two in the branching policy by which a node is chosen to branch from next. In the case of a single performance measure, branching is made at the job node having the least lower bound for the performance measure. In the current problem, two performance measures are employed. In order to decrease the computational efforts, the following policy is used here. We choose to branch the job node having the least lower bound for the primal performance measure (the number of tardy jobs) until the first feasible solution is obtained. After that, first the job nodes having the least lower bound for the number of tardy jobs are selected, and then, from these, the job node with the least lower bound for the total elapsed time is chosen for branching.

(2) Bounding procedure

In order to determine an optimal group schedule under the multiple objectives of minimizing the total elapsed time with the minimum number of tardy jobs, two kinds of lower bounds are introduced.

(a) Lower bound of the number of tardy jobs

The lower bound of the number of tardy jobs at job node N_{rs} is estimated as follows:

$$N(N_{rs}) = N_1(N_{rs}) + N_2(N_{rs}) + N_3(N_{rs}) \tag{6.10}$$

where $N_1(N_{rs})$, $N_2(N_{rs})$, and $N_3(N_{rs})$ are the numbers of tardy jobs for groups and jobs already sequenced, for jobs $(\bar{J}_r)$ not yet sequenced in $G_{(r)}$, and for groups $(\bar{G}_r)$ not yet sequenced, respectively. They are calculated in the following way:

Obviously, $N_1(N_{rs})$ is given by

$$N_1(N_{rs}) = \sum_{i=1}^{r-1}\sum_{\xi=1}^{n_i} \delta(C^K_{(i)(\xi)} - d_{(i)(\xi)}) + \sum_{\xi=1}^{s} \delta(C^K_{(i)(\xi)} - d_{(i)(\xi)}) \tag{6.11}$$

For the computation of $N_2(N_{rs})$ and $N_3(N_{rs})$, Hodgson's algorithm,[9] which gives the optimal schedule minimizing the number of tardy jobs for a single-stage scheduling, can be effectively employed. To make use of this algorithm for each stage, the hypothetical due date for operation $O^k_{(i)(\xi)}$ is defined as

$$d'^k_{(i)(\xi)} = d_{(i)(\xi)} - \sum_{h=k+1}^{K} p^h_{(i)(\xi)} \tag{6.12}$$

Then, $N^k_2(N_{rs})$, the number of tardy operations of the jobs in $\bar{J}_r$ on M_k, is estimated by applying Hodgson's algorithm.

Hence,

$$N_2(N_{rs}) = \max_{1 \le k \le K} N^k_2(N_{rs}) \tag{6.13}$$

$N^k_3(N_{rs})$, the number of tardy operations of the jobs in $\bar{G}_r$ on M_k, is calculated by the following procedure, which is an extension of Hodgson's algorithm to group scheduling.

(i) Sequence the operations of jobs in $\bar{G}_r$ for each of the stages, respectively, in order of nondecreasing hypothetical due date, irrespective of the groups to which the operations belong.

(ii) Order the group setup times and the numbers of jobs for groups not yet sequenced such that $S'^k_{(r+1)} \le S'^k_{(r+2)} \le \dots \le S'^k_{(N)}$ and such that $n'_{(r+1)} \ge n'_{(r+2)} \ge \dots \ge n'_{(N)}$, respectively. Insert each of the (N-r) group setup times, $S'^{k}_{(i)}$ ($i = r+1, r+2, \ldots, N$) into the operation order every $n'_{(i)}$ operations ($i = r+1, r+2, \ldots, N$).

(iii) Set the start time of the operation order at $C^{k}_{(r)(s)} + \sum_{\xi=s+1}^{n_r} p^{k}_{(r)(\xi)}$, and identify the first tardy operation. Suppose this to turn out to be lth operation in the order and then identify the operation with the maximum processing time among the first l operations. Remove it from the order and set $N_3^k(N_{rs}) = N_3^k(N_{rs}) + 1$. If no operations are late, then stop.

(iv) Interchange each of the group setups which are positioned after the removed operation and the operation which immediately follows the setup. Go to step (iii).

Then,

$$N_3(N_{rs}) = \max N_3^k(N_{rs}) \tag{6.14}$$

(b) Lower bound of the total elapsed time

Many kinds of lower bounds of the total elapsed time have been developed in conventional scheduling. It is well known that the composite and the revised lower bounds are effective as compared with the machine-based or the job-based lower bound. However, it is reported that the composite lower bound is not so effective in group scheduling as mentioned in Chapter 5. Hence, the revised lower bound is used here. In order to compute the revised lower bound on the total elapsed time, Theorem 4.5, which is developed for the two-stage group scheduling problem in Chapter 4, is used.

With the help of Theorem 4.5, the revised lower bound at N_{rs} is estimated as follows:

$$L(N_{rs}) = \max \left\{ \begin{array}{l} C^{1}_{(r)(s)} + \sum_{\xi=s+1}^{n_r} p^{1}_{(r)(\xi)} + \sum_{i=r+1}^{N} Q^{1}_{(i)} + \min_{i\xi\in\bar{J}_{rs}} \sum_{h=2}^{K} p^{h}_{(i)(\xi)} \\ \max_{2\le k\le K} \{C^{k}_{(r)(s)} + \sum_{\xi=s+1}^{n_r} (g^{k}_{(r)<\xi>} + p^{k}_{(r)<\xi>}) + \sum_{i=r+1}^{N} \{s^{k}_{<i>} \\ \qquad + \sum_{\xi=1}^{n_i} (g^{k}_{<i><\xi>} + p^{k}_{<i><\xi>})\} + \min_{i\xi\in\bar{J}_{rs}} \sum_{h=k+1}^{K} p^{h}_{(i)(\xi)} \end{array} \right\} \quad (6.15)$$

where $\bar{J}_{rs}$ is the set of jobs not yet sequenced at N_{rs} and the symbol < > designates the order of groups and jobs determined by applying Theorem 4.5 to each of all the two consecutive stages M_k and M_{k+1} ($k = 1, 2, \ldots, K-1$).

6.4 Determining optimal machining speeds

Once the optimal group schedule is determined, the machining speeds can be changed to reduce the total production cost if there occur slack times of some operations in the schedule.

With a decrease in machining speed, the unit production time increases and the unit production cost decreases in the high-efficiency speed range $[v^{k(c)}_{i\xi}, v^{k(t)}_{i\xi}]$.[8] All the machining speeds are initially set at the maximum-production-rate speeds in order to satisfy the minimum-total-elapsed-time constraint with the minimum number of tardy jobs. Hence, by decreasing the machining speeds from the maximum-production-rate speeds, and approaching their minimum-cost machining speeds as far as possible by utilizing the slack times, the production cost can be decreased. The problem is how to select the operation $O^{k}_{i\xi}$ for cost reduction. The following function, called an "efficiency-sensitivity function," can be employed as a measure in selecting the operations.[10]

$$\gamma^{k}_{i\xi} = -\frac{dq^{k}_{i\xi}/dv^{k}_{i\xi}}{du^{k}_{i\xi}/dv^{k}_{i\xi}}$$

$$= \frac{\dfrac{\beta^k_{i\xi}}{(v^k_{i\xi})^2} - (\dfrac{1}{n^k_{i\xi}} - 1)\varepsilon^k_{i\xi}(v^k_{i\xi})^{1/n^k_{i\xi}-2}/(C^k_{i\xi})^{1/n^k_{i\xi}}}{-\dfrac{1}{(v^k_{i\xi})^2} + (\dfrac{1}{n^k_{i\xi}} - 1)b^k_{i\xi}(v^k_{i\xi})^{1/n^k_{i\xi}-2}/(C^k_{i\xi})^{1/n^k_{i\xi}}} + \alpha^k_{i\xi} \qquad (6.16)$$

$$(i = 1, 2, \ldots, N, \ \xi = 1, 2, \ldots, n_i, \ k = 1, 2, \ldots, K)$$

This function gives an index of the amount of cost reduction over the increase of production time by decreasing the machining speed of operation $O^k_{i\xi}$. The larger the value of equation (6.16), the larger the cost reduction is for a certain amount of production time increase.

6.5 Optimizing algorithms

Based on the results of the previous analysis, the optimizing algorithms for determining an optimal group schedule and optimal machining speeds are proposed as follows:

< Optimizing algorithms for determining an optimal group schedule and optimal machining speeds >

[Stage 1] Branch-and-bound algorithm for determining an optimal group schedule.

Step 1. Set the machining speeds on all stages for all jobs at the maximum-production-rate machining speeds, $v^{k(t)}_{i\xi}$. Go to Step 2.

Step 2. Let the group level $r = 0$ and the least feasible number of tardy jobs $N^* = \infty$ and the least feasible total elapsed time $L^* = \infty$. Go to Step 3.

Step 3. Branch the group node into $(N - r)$ group node N_{r+1} by placing each of the groups not yet allocated next in the sequence determined. Set $r = r + 1$, and go to Step 4.

Step 4. For each of the group nodes N_r, create job nodes N_{rs} of the job level $s=1$ by placing each of the jobs in the group next in the sequence determined. Go to Step 5.

Step 5. Calculate the lower bound $N(N_{rs})$ for each of the new job nodes. Go to Step 6.

Step 6. Find the job node having $\min N(N_{rs})$ from among the job nodes derived in Step 4, or 9 in the case of $N^*=\infty$, or from among all job nodes being active in the case of $N^*\neq\infty$. If $N^*=\infty$ and more than two nodes having $\min N(N_{rs})$ exist, then compute $L(N_{rs})$ of these nodes, and select a job node having $\min L(N_{rs})$. (In the case of a tie, select the node with the largest value of, first, r, and then s.) Let the group level and job level of the node be r and s, respectively, and $N^*(N_{rs})=N(N_{rs})$ and $L^*(N_{rs})=L(N_{rs})$. Go to Step 7.

Step 7. If $N^*(N_{rs})>N^*$, or $N^*(N_{rs})=N^*$ and $L^*(N_{rs})\geq L^*$, then the group schedule of the node having N^* and/or L^* is optimal. Go to Stage 2. Otherwise, go to Step 8.

Step 8. If $s<n_{(r)}$, then go to Step 9. Otherwise go to Step 10.

Step 9. Branch the job node N_{rs} into $(n_{(r)}-s)$ nodes N_{rs+1} by placing each of the jobs not yet allocated in $G_{(r)}$ next in the sequence determined. Set $s=s+1$, and go back to Step 5.

Step 10. If $r<N$, then go back to Step 3. Otherwise, $N^*=N^*(N_{rs})$ and $L^*=L^*(N_{rs})$, so go back to Step 6.

[Stage 2] Algorithm for determining optimal machining speeds

Step 1. Let D denote the set of subscripts, i, ξ, and k such that operations are not critical under the optimal group schedule determined. Compute the efficiency-sensitivity functions for $O_{i\xi}^{k}$ $(i\xi k\in D)$ as follows:

$$\hat{\gamma}_{i\xi}^{k}=\gamma_{i\xi}^{k}\Big|_{v_{i\xi}^{k(t)}-\Delta v}$$

where Δv (>0) is a small speed value. Go to Step 2.

Step 2. $\gamma=\max\limits_{i\xi k\in D}\hat{\gamma}_{i\xi}^{k}$. Let U denote the set of subscripts i, ξ, and k such that $\gamma = \max \hat{\gamma}^k_{i\xi}$. $\gamma = \gamma - \Delta\gamma$, where $\Delta\gamma$ (> 0) is a small value. Go to Step 3.

Step 3. $U = U + \{i_o \xi_o k_o\}$, where $\{i_o \xi_o k_o\}$ is the set of subscripts such that $\gamma \leq \hat{\gamma}^k_{i\xi}$. Go to Step 4.

Step 4. (i) For $0^k_{i\xi}$ ($i\xi k \in U$), compute the machining speeds $\hat{v}^k_{i\xi}$ and the job processing times $\hat{p}^k_{i\xi}$ such that $\gamma = \hat{\gamma}^k_{i\xi}$.

(ii) Calculate the slack times $t^k_{si\xi}$ for $0^k_{i\xi}$ ($i\xi k \in U$) as follows:

$$t^k_{si\xi} = t^k_{li\xi} - t^k_{ei\xi} - \hat{p}^k_{i\xi}$$

where $t^k_{ei\xi}$ and $t^k_{li\xi}$ are the earliest starting time and the latest finishing time, respectively, under the minimum-total-elapsed-time schedule with the minimum number of tardy jobs.

(iii) If there exists any operation $0^k_{i\xi}$ ($i\xi k \in U$) such that $t^k_{si\xi} < 0$, then $\gamma = \gamma + \Delta\gamma'$ (for example $\Delta\gamma' = \frac{1}{2}\Delta\gamma$) and go back to Step 3. Otherwise, for $\hat{0}^{k_o}_{i_o\xi_o}$ such that $t^{k_o}_{si_o\xi_o} = 0$, the optimal machining speeds $v^{*k_o}_{i_o\xi_o}$ are given by $\hat{v}^{k_o}_{i_o\xi_o}$. $U = U - \{i_o \xi_o k_o\}$, $D = D - \{i_o \xi_o k_o\}$.

(iv) If $D \neq \phi$, go to Step 5. Otherwise, Stop.

Step 5. If $\gamma = 0$, the optimal machining speeds $v^{*k}_{i\xi}$ for $0^k_{i\xi}$ ($i\xi k \in U$) are given by $v^{k(c)}_{i\xi}$. Stop. If $\gamma \neq 0$, then $\gamma = \gamma - \Delta\gamma$, so go back to Step 3.

6.6 Numerical examples

In order to verify the effeciveness of the proposed optimizing algorithms for determining the optimal group schedule and the optimal machining speeds, a hypothetical example is presented below.

Eight kinds of shafts are to be processed in a lot size of five on a flow-type, four-stage manufacturing system (rough machining, finishing machining,grooving, and threading). These shafts are classified into three groups according to their dimensions. Fig. 6.1 shows sketches of representative parts for each of the three groups. Production data for each job (shaft) of the three groups are given in Tables 6.1, 6.2, and 6.3.

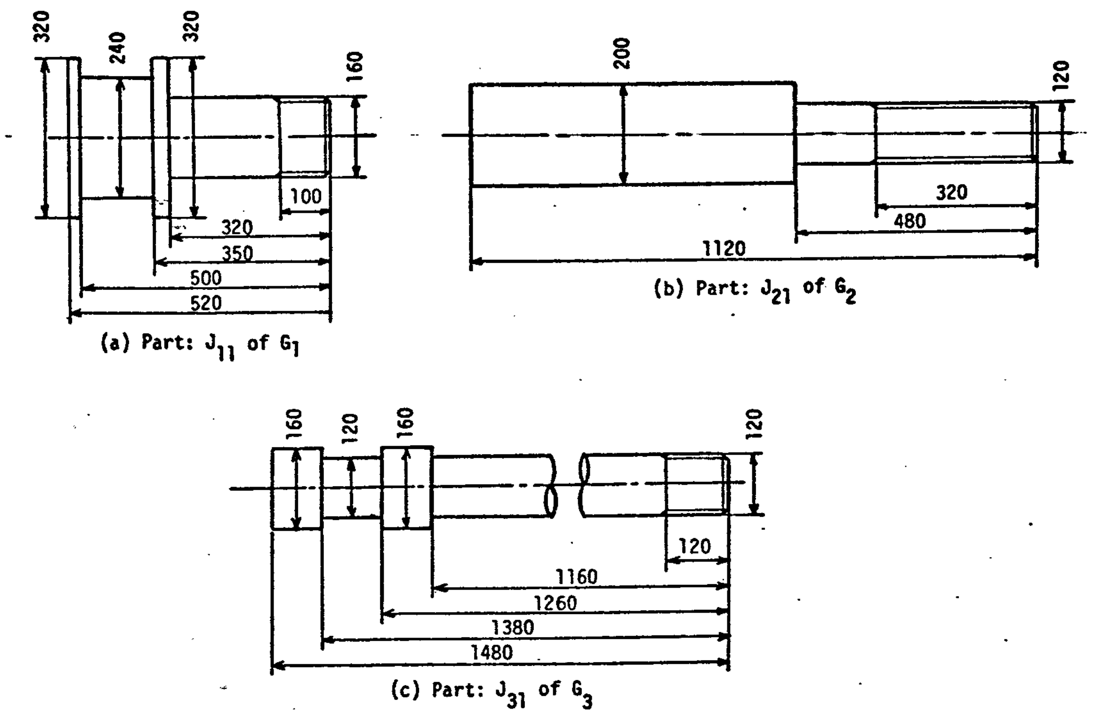


Fig. 6.1 Sketches of representative parts for each of three groups

Table 6.1 Group setup times

(units:min)

Stage \ Group	G_1	G_2	G_3
M_1	10.00	9.00	7.00
M_2	12.00	10.00	9.00
M_3	8.00	7.00	11.00
M_4	11.00	12.00	8.00

Table 6.2 Jobs'due dates

(units: min)

Group	G_i	G_1		G_2			G_3		
Job	$J_{i\xi}$	J_{11}	J_{12}	J_{21}	J_{22}	J_{23}	J_{31}	J_{32}	J_{33}
Due date	$d_{i\xi}$	930	450	1080	450	830	650	860	510

Table 6.3 Production data for jobs to be machined

			Processing parameter						Tool parameter		Time parameter			Cost parameter		
Group	Job	Stage	Depth of cut	Feed Rate	Work Diameter		Work Length		1-min tool life speed	Slope constant	Job setup time	Prepa-ration time	Tool exchange time	Direct labor cost and overhead	Machining overhead cost	Tool cost
G_i	$J_{i\xi}$	M_k	$g^k_{i\xi}$ (min)	$f^k_{i\xi}$ (mm/rev)	$D^k_{i\xi}$ (mm)	$D^k_{i\xi}$ (mm)	$L^k_{i\xi}$ (mm)	$L'^k_{i\xi}$ (mm)	$C^k_{i\xi}$ (m/min)	$n^k_{i\xi}$	$s^k_{i\xi}$ (min)	$a^k_{i\xi}$ (min/pc)	$b^k_{i\xi}$ (min/edge)	$\alpha^k_{i\xi}$ ($/min)	$\beta^k_{i\xi}$ ($/min)	$\varepsilon^k_{i\xi}$ ($/min)
G_1	J_{11}	M_1	1.50	0.25	320	120	200	320	340	0.20	2.00	2.50	2.50	0.35	0.15	9.00
		M_2	0.50	0.15	320	120	200	320	250	0.25	1.50	3.00	3.00	0.45	0.30	12.50
		M_3	1.00	0.10	240	—	150	—	300	0.33	1.50	5.00	1.50	0.35	0.15	10.00
		M_4	1.00	0.26	160	—	100	—	350	0.20	3.00	4.00	3.50	0.35	0.15	9.50
	J_{12}	M_1	1.00	0.25	280	120	360	280	350	0.25	0.60	3.00	4.00	0.35	0.20	9.50
		M_2	0.50	0.12	280	—	360	—	200	0.20	0.70	2.50	4.50	0.45	0.25	12.00
		M_3	1.00	0.15	200	—	120	—	320	0.33	0.80	3.50	3.50	0.35	0.10	13.00
		M_4	—	—	—	—	—	—	—	—	—	—	—	—	—	—
G_2	J_{21}	M_1	1.50	0.30	200	120	640	480	400	0.20	1.40	3.00	5.00	0.35	0.40	11.50
		M_2	0.50	0.12	200	—	640	—	300	0.28	0.90	3.50	3.50	0.45	0.35	12.50
		M_3	—	—	—	—	—	—	—	—	—	—	—	—	—	—
		M_4	1.50	0.35	120	—	320	—	350	0.20	2.00	5.00	4.00	0.35	0.25	14.00
	J_{22}	M_1	1.50	0.25	280	200	680	400	300	0.23	2.00	4.00	5.00	0.35	0.35	9.00
		M_2	0.50	0.15	280	—	680	—	250	0.33	0.70	3.50	2.00	0.45	0.25	14.00
		M_3	—	—	—	—	—	—	—	—	—	—	—	—	—	—
		M_4	1.00	0.30	200	—	200	—	270	0.23	2.00	2.00	3.00	0.35	0.25	15.00
	J_{23}	M_1	1.50	0.25	320	120	560	520	350	0.20	3.00	3.50	5.00	0.35	0.20	12.00
		M_2	0.50	0.15	320	—	560	—	200	0.25	2.20	4.00	3.00	0.45	0.25	13.50
		M_3	1.00	0.20	240	—	160	—	200	0.20	0.00	2.00	3.00	0.35	0.25	14.50
		M_4	1.00	0.25	120	—	200	—	250	0.22	5.00	3.00	4.00	0.35	0.10	9.00
G_3	J_{31}	M_1	2.00	0.25	160	120	320	1160	400	0.25	1.00	4.00	3.50	0.35	0.12	6.50
		M_2	0.50	0.15	160	—	320	—	250	0.20	0.60	3.00	4.00	0.45	0.30	13.50
		M_3	1.00	0.20	120	—	120	—	300	0.33	1.00	3.50	2.00	0.35	0.20	12.50
		M_4	1.00	0.30	120	—	120	—	320	0.20	1.50	2.50	3.00	0.35	0.25	14.00
	J_{32}	M_1	1.50	0.25	200	120	320	1320	500	0.25	1.00	5.00	2.50	0.35	0.15	7.50
		M_2	0.50	0.12	200	—	320	—	400	0.20	0.80	4.50	4.00	0.45	0.25	12.50
		M_3	—	—	—	—	—	—	—	—	—	—	—	—	—	—
		M_4	1.00	0.30	120	—	120	—	280	0.33	2.30	5.00	2.00	0.35	0.40	13.00
	J_{33}	M_1	2.00	0.20	240	200	500	900	300	0.20	1.30	5.00	2.50	0.35	0.15	7.50
		M_2	0.50	0.15	240	200	500	900	340	0.25	1.00	3.50	3.00	0.45	0.32	9.50
		M_3	1.00	0.20	200	—	150	—	250	0.33	1.20	4.50	4.00	0.35	0.20	12.00
		M_4	1.50	0.30	200	—	150	—	400	0.20	3.00	3.00	3.50	0.35	0.35	10.00

Table 6.4 Job processing times at the maximum-production-rate machining speeds

(units: min)

Group	G_1		G_2			G_3		
Job	J_{11}	J_{12}	J_{21}	J_{22}	J_{23}	J_{31}	J_{32}	J_{33}
Stage M_1	51.99	75.47	71.69	150.88	119.20	92.78	87.67	181.90
Stage M_2	115.56	160.21	161.91	207.22	238.89	62.28	68.88	231.89
Stage M_3	62.61	40.69	—	—	40.98	27.43	—	51.78
Stage M_4	27.35	—	37.72	29.13	33.86	18.84	33.68	26.88

The machining constant, $\lambda^k_{i\xi}$, is calculated as a function of the diameter, length, and feed rate.[11)]

Job processing times used for group scheduling are determined by

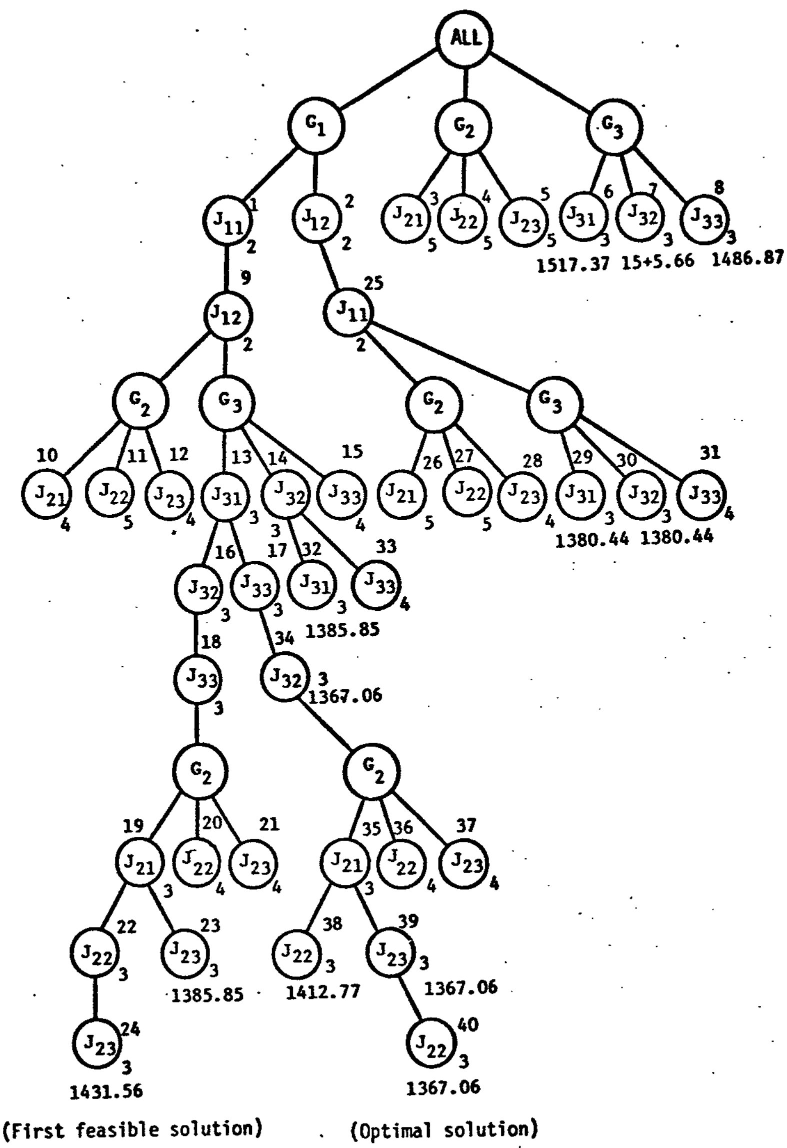


Fig. 6.2 The branching tree for the example problem

setting machining speeds at the maximum-production-rate machining speeds as shown in Table 6.4. The optimal group schedule is determined by the optimizing algorithm based on the branch-and-bound method. The branching tree for this problem is displayed in Fig. 6.2. The lower bound of the number of tardy jobs for each subproblem is given just below the corresponding node in the figure. The order of branching is indicated by the number that appears just above the corresponding node. The first feasible solution is obtained at node 24, which gives the group schedule $G_1(J_{11}-J_{12})-G_3(J_{31}-J_{32}-J_{33})-G_2(J_{21}-J_{22}-J_{23})$. Then, the lower bound of the total elapsed time, which is indicated just below the number of tardy jobs, is calculated. The optimal solution is $G_1(J_{11}-J_{12})-G_3(J_{31}-J_{33}-J_{32})-G_2(J_{21}-J_{23}-J_{22})$, which has 3 as the number of tardy jobs and the total elapsed time of 1367.06 min (see Table 6.5).

In this schedule, there are slack times available to reduce the total production cost as shown in Table 6.5. Then, the optimal machining speed for each operation is determined by the optimizing algorithm as shown in Table 6.6. The job processing times and costs at the machining speeds are also given in Table 6.6. The total production cost is reduced to \$2949.20 from \$3191.25.

Table 6.5 Job completion times, job tardinesses, and slack times for operations under the optimal group schedule

(units: min)

Optimal group sequence		G_1		G_3			G_2		
Optimal job sequence		J_{11}	J_{12}	J_{31}	J_{33}	J_{32}	J_{21}	J_{23}	J_{22}
Job completion time $C_{i\xi}$		267.51	378.45	455.31	729.69	763.37	929.54	1205.55	1367.06
Job tardiness $T_{i\xi}$		0	0	0	219.69	0	0	375.55	917.06
Slack time for operation $t^k_{si\xi}$	Stage M_1	0	0	0	0	144.22	132.41	185.12	273.13
	Stage M_2	10.10	10.10	10.10	0	0	0	0	0
	Stage M_3	394.53	71.55	194.69	0	—	—	0	—
	Stage M_4	443.34	—	194.69	0	96.63	150.46	0	0

Table 6.6 Optimal machining speeds, job processing times and costs

Group	Job	Stage	Optimal machining speed	Job production time	Job production cost
G_i	$J_{i\xi}$	M_k	$v^{*k}_{i\xi}$ (m/min)	$p^{*k}_{i\xi}$ (min)	$q^{*k}_{i\xi} \times l_{i\xi}$ ($)
G_1	J_{11}	M_1	215^t	51.99	49.00
		M_2	118^e	120.74	76.09
		M_3	87^c	86.38	51.90
		M_4	144^c	28.20	10.10
	J_{12}	M_1	188^t	75.47	70.75
		M_2	100^e	163.56	156.10
		M_3	81^c	51.00	26.95
		M_4	—	—	—
G_2	J_{21}	M_1	171^c	77.36	58.60
		M_2	162^t	161.91	252.15
		M_3	—	—	—
		M_4	139^c	39.92	18.10
	J_{22}	M_1	128^c	164.37	106.80
		M_2	157^t	207.22	567.55
		M_3	—	—	—
		M_4	159^t	29.13	37.90
	J_{23}	M_1	139^c	134.89	81.00
		M_2	115^t	238.89	390.90
		M_3	122^t	40.98	50.50
		M_4	139^t	33.86	18.05
G_3	J_{31}	M_1	222^t	92.78	71.90
		M_2	126^e	63.74	59.65
		M_3	83^c	31.65	17.30
		M_4	127^c	20.10	8.80
	J_{32}	M_1	188^c	104.11	58.35
		M_2	230^t	68.88	68.25
		M_3	—	—	—
		M_4	85^c	36.65	18.70
	J_{33}	M_1	189^t	181.90	175.25
		M_2	196^t	231 89	324.05
		M_3	125^t	50.58	49.25
		M_4	236^t	26.88	15.90

Note: The symbols t, c, and e above the machining speeds indicate the maximum-production-rate machining speeds, the minimum-cost machining speeds, and the machining speeds in the high-efficiency speed range, respectively.

6.7 Conclusions

(1) The group scheduling model was constructed on the multiple production stages with variable processing times and costs depending on machining

conditions.

(2) The optimal group schedule for the minimum total elapsed time with the minimum number of tardy jobs was determined by the branch-and-bound algorithm.

(3) The optimal machining speeds minimizing the total production cost were determined by utilizing the slack times under the optimal schedule.

(4) The effectiveness of the optimizing algorithms for determining the optimal group schedule and the optimal machining speeds were tested with numerical examples.

CHAPTER 7 EXPERIMENTAL INVESTIGATION OF GROUP PRODUCTION SCHEDULING

7.1 Introduction

In the previous chapters, group scheduling models have been developed under static conditions where jobs are available simultaneously for processing. In real situations, however, jobs to be processed often arrive at the shop randomly over time. Jobs are assigned to each of the machines according to their processing routes. In this case, scheduling is generally carried out by means of dispatching decisions. This chapter deals with group scheduling under these dynamic conditions.

Group scheduling that differs from conventional scheduling has some specific features as follows:

(1) Jobs to be processed are to belong to one of the setup groups which are classified according to a classification and coding system by GT.

(2) Since the setup time and the setup group play a critical role in group scheduling, scheduling rules, including the setup time and the setup group of the job in the queue, are to be investigated in a simulation run.

(3) With group technology, the job flow is expected to be a flow-shop pattern or a near-flow-shop pattern; hence, in order to investigate the group scheduling under dynamic conditions, simulation experiments are to be run for the flow-shop pattern or the near-flow-shop pattern rather than the job-shop pattern.

So far, much research on conventional scheduling has been done to study a large variety of scheduling (dispatching or priority) rules by which the jobs in the queue are assigned to the idle machines in the job shop,[1~5] and over 100 such rules have been reported.[6] However there have been very few studies on the scheduling rules in the near flow shop and the flow shop.[7] In group scheduling, setup times play an important role as mentioned above. Therefore, the influences of setup times on shop performances should be investigated in order to clarify the feature of the group scheduling under dynamic conditions. However, only few studies have been made on this subject.[8,9]

In this chapter, the effect of the types of flow patterns — job-shop, near-flow-shop, and flow-shop patterns on the measures of performance and the scheduling rules is firstly investigated for group scheduling.[10]

It is well known that the shop load greatly influences the comparative performance of the scheduling rules for conventional scheduling. In the case of group scheduling, it can be expected that the relative size of the setup time to the processing time and the variance of the setup time also influence the shop performances. In order to study the effects of each of these three factors on each individual scheduling rule, analysis of variance is performed on each of the measures of performance in the flow shop.

Then the effect of the relative size of the setup time to the processing time on the goodness of the four scheduling rules is investigated for the three flow patterns.

In comparing scheduling rules through scheduling simulation, usually the processing times of jobs are assumed to be random variables generated from an exponential distribution. In addition, the processing times are obtained from a normal distribution in an attempt to investigate the effect of the differences in the distributions.[11]

7.2 Simulation model for group scheduling

7.2.1 Group scheduling model under dynamic conditions

In an attempt to construct a simulation model of group scheduling under dynamic conditions the following preconditions are set:

(i) Jobs to be processed are classified into several setup groups.

(ii) Job processing times requiring completion of jobs consist of setup times and production times (machining times)

(iii) Setup times are dependent on the sequence of the groups to which jobs belong.

7.2.2 Assumptions for simulation model

A simulation model for group scheduling is constructed under the following assumptions:

(i) Each machine is continuously available for assignment without intermittent unavailability.

(ii) Jobs are simple sequences of operations.

(iii) Each operation can be processed by only one machine.

(iv) No preemption and no overlap scheduling.

(v) Each machine can handle at most one operation at a time.

(vi) Instantaneous transfer to next machine after completion of an operation.

(vii) The job arrivals follow a Poisson process.

(viii) The production times are random variables obtained from an exponential distribution or a normal distribution. The setup-time distribution is assumed to be a uniform form.

(ix) Information on the production times and the setup times is available for the scheduling procedures.

7.2.3 Types of flow patterns

In order to investigate the effect of the types of flow patterns on the measures of performance, simulation experiments are run in the following shops.

(1) Job shop (Flow pattern F1) in which there is no common pattern of movement of the jobs from one machine to another.

(2) Flow shop (Flow pattern F3) in which all the jobs flow essentially the same path from one machine to another.

(3) Intermediate shop or near flow shop (Flow pattern F2) which falls somewhere between the job shop and the flow shop. The flow patterns of the jobs in this shop depend on their setup groups.

In the above three shops, all the jobs processed have the same number of operations. In addition to the above shops, two shops — job shop (Flow pattern F4) and flow shop (Flow pattern F5) are set in order to investigate the effect of the difference in the mean number of operations of a job on the measures of performance. In these shops, the number of operations of a job is a random variable obtained from a uniform distribution.

7.2.4 Measures of performance

The measure of performance employed in the simulation model is the mean flow time. This measure is a reasonable choice, since a rule which minimizes the mean flow time will also minimize the mean waiting time and the mean number of the jobs in the queue.

In addition to this measure, the maximum flow time is employed as a secondary measure.

7.2.5 Scheduling rules

Eight scheduling rules for giving priorities to the jobs in the queue are tested in this simulation study; three of them involve setup times.

(1) RANDOM (Random): Job is chosen from the queue on a random basis with no consideration given to job characteristics.

(2) FCFS (First-Come, First-Served): Jobs are removed from the queue in the same order as they entered.

(3) SMT (Shortest Machining Time): Job with the shortest machining time has priority.

(4) SPT (Shortest Processing Time): Job with the shortest processing

time is given priority.

(5) LWKR (Least Work Remaining): Job with the least sum of the mean processing times for all operations not yet performed has priority.

(6) FOPNR (Fewest Operations Remaining): Job with the fewest number of operations remaining to be performed on the job has priority.

(7) SST (Shortest Setup Time): Job with the shortest setup time has priority.

(8) TSS (Traveling Salesman Sequence): In this rule, all the jobs of a given group in the queue are processed and then the jobs of another group are processed in a fixed sequence which minimizes the total setup time in a full cycle. This sequence is given by the solution of the traveling salesman problem for the setup-time matrix. If the group which normally follows in the sequence is not represented in the queue, it is disregarded, and the job of the next group in the sequence is processed. Within the group having priority, the job with the shortest machining time is selected.

In the simulation model for group scheduling, the SST and TSS rules which include the setup time and the setup group can be expected to play an important role in the measures of performance.

These setup-time oriented rules are evaluated in comparison with other well-known scheduling rules, such as SPT and LWKR in the latter part of the simulation experiments.

7.2.6 Parameters of model

Three parameters are defined to run the simulation experiments for group scheduling:

(1) Shop load (L); defined as the ratio of the mean processing time per job to the product of the number of machines and the mean interarrival interval.

(2) Setup time ratio (R); defined as the ratio of the mean setup time to the mean processing time.

(3) Setup time variance (V); defined as the ratio of the range of the setup time to the mean setup time.

In the simulation experiments, the values of the above three parameters are specified. Then the random variables for the interarrival intervals of the jobs, the production times and the setup times of the operations are generated according to the parameter values of the distributions obtained from the given values of the shop load, the setup time ratio, and the setup time variance.

7.3 Experimental design for simulation

The conditions of the simulation model for group scheduling are set as follows:

(i) The number of machines in each of the five shops is set at six.

(ii) The number of operations for a job is six in flow patterns, F1, F2, and F3; and the number of operations for a job is uniformly distributed from 1 to 6 operations in flow patterns, F4 and F5.

(iii) The mean processing time per job is set at 6 hours in each of the five flow patterns.

(iv) The number of setup groups is set at eight. The flow pattern of the job in each setup group for the near flow shop is shown in Table 7.1. An example of the setup-time matrix of the eight groups is shown in Table 7.2.

(v) The jobs are equally likely to belong to any one group.

The experiments were run to investigate the effects of the types of flow patterns, the parameters, the setup time ratio, and the differences in distributions. The experimental conditions for each case were as follows:

Table 7.1 Flow pattern of near flow shop

Operation No.	1	2	3	4	5	6
Group 1	M_1	M_3	M_2	M_4	M_6	M_5
Group 2	M_2	M_3	M_1	M_5	M_6	M_4
Group 3	M_1	M_2	M_5	M_3	M_4	M_6
Group 4	M_3	M_2	M_1	M_6	M_4	M_5
Group 5	M_2	M_1	M_5	M_3	M_6	M_4
Group 6	M_1	M_4	M_3	M_2	M_6	M_5
Group 7	M_3	M_4	M_2	M_1	M_5	M_6
Group 8	M_2	M_1	M_3	M_6	M_5	M_4

Note: M_k represents machine k.

Table 7.2 Setup-time matrix

$i \backslash j$	1	2	3	4	5	6	7	8
1	0.00	0.20	0.17	0.17	0.15	0.25	0.19	0.21
2	0.11	0.00	0.19	0.24	0.19	0.20	0.17	0.12
3	0.23	0.17	0.00	0.25	0.10	0.12	0.18	0.13
4	0 09	0.12	0.15	0.00	0.13	0.19	0.24	0.15
5	0.24	0.25	0.25	0.22	0.00	0.25	0.14	0.13
6	0.21	0.21	0.25	0.20	0.13	0.00	0.12	0.10
7	0.21	0.24	0.14	0.20	0.10	0.10	0.00	0.14
8	0.10	0.14	0.10	0.09	0.12	0.25	0.14	0.00

Note: The (i,j) element is the setup time required for the machine to start a group G_j job, having just processed a group G_i job. The fixed sequence which minimizes the total setup time in a full cycle is G_1-G_3-G_5-G_7-G_6-G_8-G_4-G_2-G_1. This setup-time matrix is an example for a machine in the job shop F1. (L = 0.95, R = 0.15, V = 1.00.)

(1) Effect of types of flow patterns

The simulation experiments are run five times for each of the scheduling rules in each of the five shops in order to investigate the effect of the types of flow patterns on the measures of performance, such as the mean flow time and the maximum flow time. In the simulation runs, the effect of the length of operations of the jobs on the performances is also studied. The values of the three parameters— the shop load, the setup time ratio and the setup time variance, are set at $L = 0.95$, $R = 0.15$, and $V = 1.00$, respectively.

(2) Effect of parameters

In an attempt to investigate the effects of each of the three parameters on each of the four scheduling rules, SPT, LWKR, SST, and TSS, for the flow pattern, F5, a $3 \times 3 \times 3$ factorial experimental design is used with factor levels defined as follows:

Factor L: Level of shop load

L_1: Heavy load ($L_1 = 0.95$)

L_2: Medium load ($L_2 = 0.80$)

L_3: Light load ($L_3 = 0.65$)

Factor R: Level of setup time ratio

R_1: Small setup time ratio ($R_1 = 0.05$)

R_2: Medium setup time ratio ($R_2 = 0.15$)

R_3: Large setup time ratio ($R_3 = 0.30$)

Factor V: Level of setup time variance

V_1: Small setup time variance ($V_1 = 0.50$)

V_2: Medium setup time variance ($V_2 = 1.00$)

V_3: Large setup time variance ($V_3 = 2.00$)

In the experiments, the best estimates of within-cell variance (variance for each of the combinations of factors) for each of the performance measures for each of the four scheduling rules are computed by the analysis of variance technique.

(3) Effect of setup time ratio

The simulation experiments are run for four different levels of the setup time ratio —— R = 0.05, 0.15, 0.30, and 0.50, in order to investigate the effect of the setup time ratio on the relative goodness for the four scheduling rules for each of the flow patterns, F1, F2, and F3. In the runs, other parameters are set at L = 0.95 and V = 1.00.

(4) Effect of difference in distributions

In an attempt to investigate the effect of the difference in distributions by which production times are generated, the simulation experiments are run for two kinds of production times obtained from an exponential distribution and a normal distribution. These runs are done for flow pattern F5, setting parameters L = 0.95 and V = 1.00.

7.4 Experimental results

In order to get the steady-state condition, data on jobs numbering 301~1300 (on the last 1000 jobs) were collected for each run. Based on the experimental results, the following points are noted:

(1) Effect of types of flow patterns

The experimental results of the simulation runs for the five flow patterns are given in Tables 7.3 and 7.4.

A. Mean flow time

(i) The experimental results for the flow pattern, F4, which has been used for the conventional scheduling simulations, are coincident with the results in the previous studies by other researchers; that is, the rankings of the FCFS, SPT, FOPNR, and LWKR rules were the same as rankings documented in the studies over the mean flow time and its

Table 7.3 Means and standard deviations of mean flow times

Rules	Flow pattern				
	F1	F2	F3	F4	F5
RAMDOM	82.90 28.30	78.27 22.62	85.16 30.41	96.98 34.55	105.56 27.00
FCFS	87.29 32.60	80.25 22.68	86.48 27.46	102.79 33.56	110.59 24.22
SMT	31.33 6.50	31.70 6.33	30.92 6.75	28.06 4.92	26.57 2.11
SPT	30.22 6.66	30.39 5.35	29.91 6.78	27.62 5.02	26.48 2.03
FOPNR	67.01 14.48	47.20 9.40	29.91 6.78	53.78 9.13	31.02 1.18
LWKR	46.87 6.88	46.03 7.01	38.97 7.87	39.09 5.06	28.73 3.68
SST	34.02 6.32	35.89 6.82	32.43 5.88	43.63 12.05	44.71 9.47
TSS	38.09 6.84	40.92 8.75	34.46 6.19	50.14 13.30	49.04 11.11

Note: Upper value: mean (hours), Lower value: standard deviation; Parameters: $L = 0.95$, $R = 0.15$, $V = 1.00$.

Table 7.4 Means and standard deviations of maximum flow times

Rules	Flow pattern				
	F1	F2	F3	F4	F5
RANDOM	301.5 98.3	289.2 87.4	324.0 120.4	664.1 220.8	720.4 185.7
FCFS	124.5 20.7	144.3 39.2	120.1 30.7	282.5 67.9	260.0 39.5
SMT	545.9 181.3	552.9 264.8	582.5 327.5	667.7 203.2	660.3 201.3
SPT	576.8 233.5	543.3 264.0	589.9 373.8	634.9 233.5	646.5 180.9
FOPNR	373.3 250.0	468.8 215.1	589.9 373.8	908.6 215.9	773.2 119.6
LWKR	748.8 132.0	648.6 171.4	824.2 180.5	797.6 188.4	741.3 286.7
SST	124.1 28.7	143.5 28.5	140.5 26.1	266.4 85.1	262.9 75.3
TSS	154.1 18.6	155.9 21.0	151.0 32.4	273.3 53.1	295.1 79.0

Note: Upper value: mean (hours), Lower value: standard deviation; Parameters: $L = 0.95$, $R = 0.15$, $V = 1.00$.

variance, respectively. Thus, the conventional scheduling rules appear to have the same effect on the mean flow time even in group scheduling.

(ii) There was no statistical difference in the mean flow time for the five flow patterns investigated even at the 95% confidence level for any scheduling rule except FOPNR and LWKR. Irrespective of the types of flow patterns, the mean flow time was minimal for the SPT rule, and it increased in the order of SMT, SST, and then TSS. It is interesting to note that the SPT rule performed best even in flow shops F3 and F5, since intuitively it is expected that LWKR and FOPNR are superior to SPT for the flow shops.

(iii) As expected, the FOPNR and LWKR rules performed better in the flow-shop patterns than in the job-shop patterns because the jobs could leave the flow shops quickly by processing the jobs with the least remaining works. The differences, except in the LWKR values for F1 and F3, were statistically significant.

(iv) In order to investigate the effect of the length of operations, the performances for the six-fixed length of operations flow types (F1, F2, F3) were compared to those for the variant length of operations flow types (mean operation length is 3.5) (F4, F5). Due to unknown reasons the mean flow time for each of the rules, RANDOM, FCFS, SST, and TSS, showed a smaller value for the fixed-operations flow types than that in the variant-operations flow types. While, the SMT and SPT values were less for the variant-operations flow types than for the fixed-operations flow types. This result indicates that the relative effectiveness of the scheduling rules on the mean flow time is larger for the variant-operations flow types than for the fixed-operations flow types. However, the differences were not significant.

B. Maximum flow time

(i) The maximum flow time did not show any statistical difference for the fixed-operations flow types for any rule. There was also no significant difference in the maximum flow time for the variant-operations flow types.

(ii) It is interesting to note that the maximum flow times for the variant-operations flow types showed significantly larger values than those for the fixed operations flow types since it is expected that there is no difference for the two flow types as is the case with the mean flow time.

(iii) The FCFS and SST rules performed best and were followed by TSS for all flow patterns.

(2) Effect of parameters

The results of analysis-of-variance computations are given in Tables 7.5 and 7.6. The numbers entered in the columns of the tables represent the percentages of variance caused by the source factors listed in the stub column.

Table 7.5 Results on percentage of mean flow time

Source factor	Percentage variance for the rule			
	SPT	LWKR	SST	TSS
Shop load (L)	96.76	97.43	50.62	51.87
Setup time ratio (R)	2.70	1.06	26.02	25.22
Setup time variance (V)	0.08	0.10	0.15	0.00
Interaction of L and R	0.12	0.76	23.03	22.76
Interaction of R and V	0.06	0.25	0.04	0.06
Interaction of L and V	0.15	0.18	0.05	0.01
Error	0.13	0.22	0.09	0.08
Grand mean	18.36	21.48	29.64	33.05
Standard deviation	5.20	6.17	21.50	24.58

Table 7.6 Results on percentage of maximum flow time

Source factor	Percentage variance for the rule			
	SPT	LWKR	SST	TSS
Shop load (L)	93.52	83.06	47.19	40.68
Setup time ratio (R)	0.41	1.12	29.41	35.02
Setup time variance (V)	0.18	0.29	0.83	0.43
Interaction of L and R	1.58	7.78	20.20	19.02
Interaction of R and V	0.46	2.71	0.16	0.18
Interaction of L and V	1.41	1.41	1.23	2.45
Error	2.44	3.63	0.98	2.21
Grand mean	344.07	578.52	197.11	204.11
Standard deviation	217.25	265.52	128.54	119.55

The results showed the following points.

A. Mean flow time

(i) As expected, the load factor was most prominent and caused about 97% of the variance for the SPT and LWKR rules.

(ii) For the SST and TSS rules, the load factor had the largest effect of about 50%. The setup time ratio factor was next in importance and the load and setup time ratio interaction exerted an almost equal influence on this performance. This result indicates that the goodness of the SST and TSS rules depends on the setup time ratio.

(iii) Against expectation, the setup time variance had little influence on the mean flow time for all rules. This result may be explained by the fact that the setup-time distribution was assumed a uniform form.

B. Maximum flow time

In regards to within-cell variance, almost the same results as for the mean flow time were obtained.

(3) Effect of setup time ratio

The effects of the setup time ratio on the measures of performances are shown in Figures 7.1 and 7.2 for the three flow patterns. The scheduling rules employed in the experiments were SPT, LWKR, SST, and TSS.

(i) With small values of the setup time ratio, the SPT rule showed the smallest mean flow time, LWKR and SST the next smaller ones, and TSS the largest one.

(ii) As the setup time ratio increased, the SST and TSS values of the mean flow times decreased and became equal to or less than the SPT value.

(iii) The LWKR and SPT rules provided a bad performance on the maximum flow time for all the setup time ratios. For the larger setup time ratios, such as 0.30 and 0.50, the SST and TSS rules performed slightly better than the SPT rule on the mean flow time, while the differences were not significant. Considering that SST and TSS performed best on the maximum flow time, it is concluded that SST and TSS apppear to dominate SPT for the large setup time ratios.

(4) Effect of difference in distributions

Figures 7.3 and 7.4 show the effect of the difference in the production-time distributions. It seems from these figures that there is no large difference in superiority of each of the scheduling rules and in performance values for the two kinds of distributions. In the case in which the production times are obtained from a normal distribution, there is a tendency to decrease the maximum flow time for SPT when the setup time ratio is as large as 0.50. This is explained by the fact that the standard deviation is as small a value as half the mean, and hence the maximum flow time is dominated

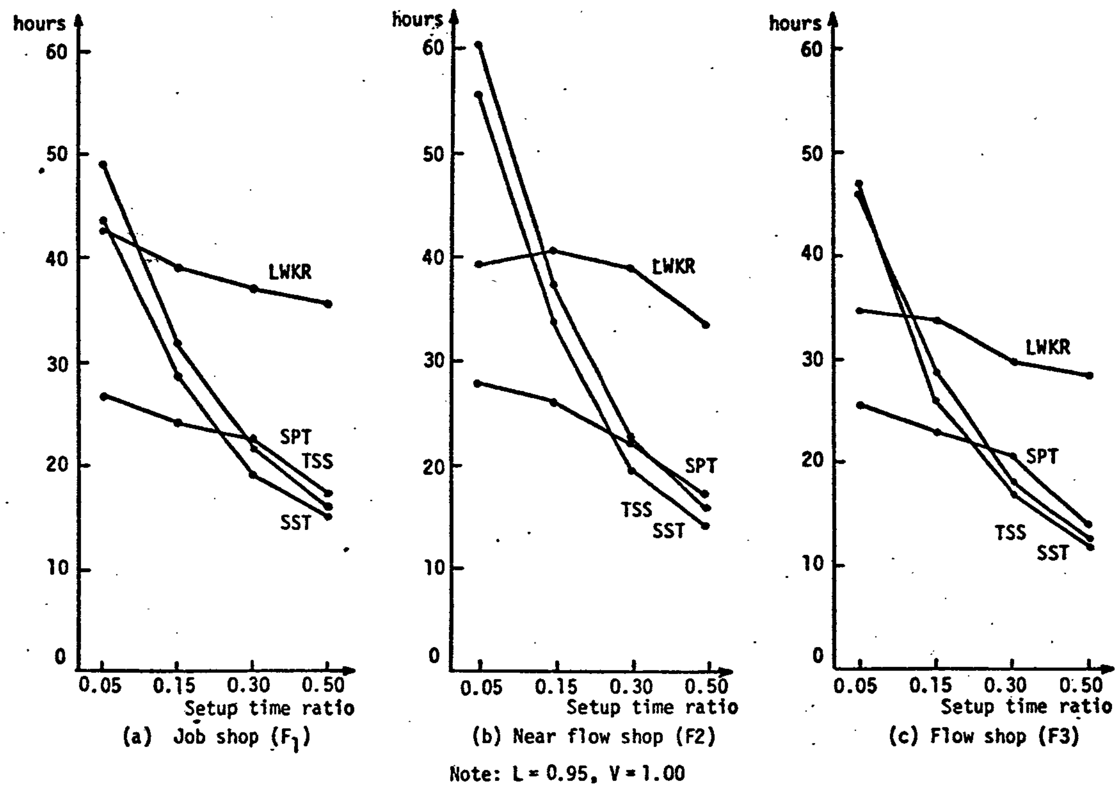


Fig. 7.1 Mean flow times for three flow patterns

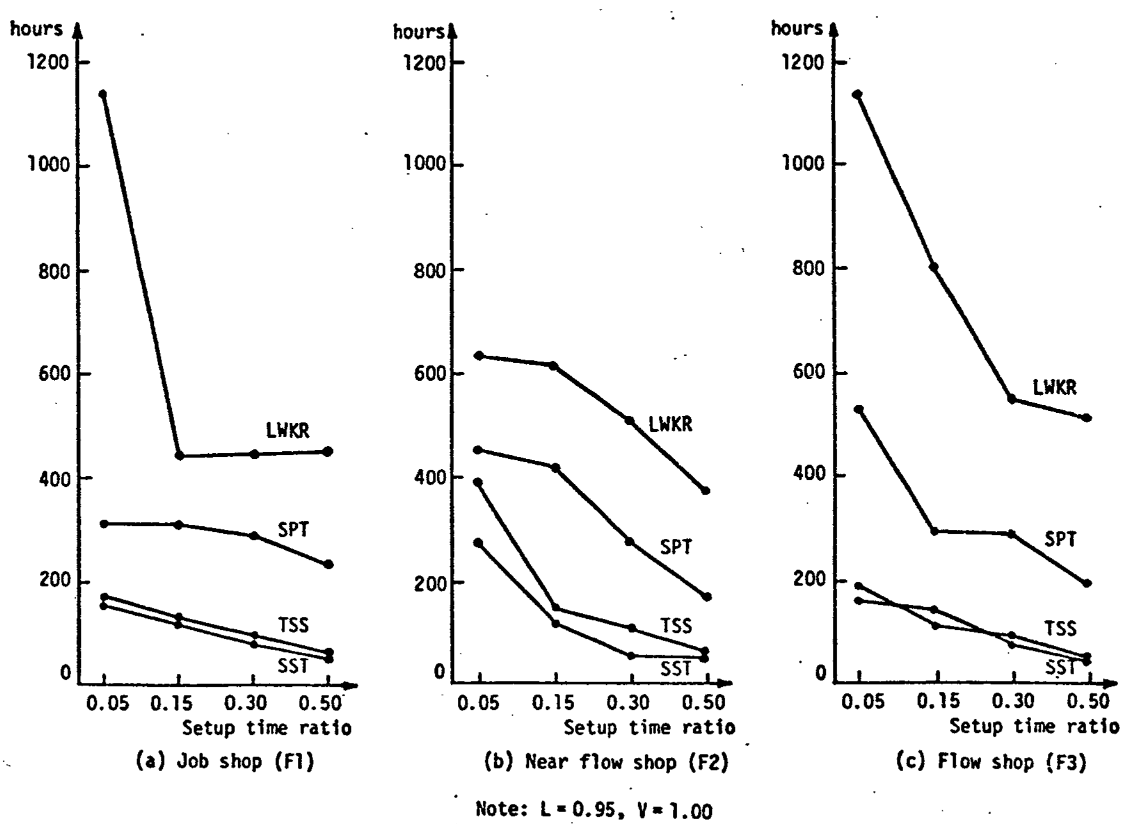


Fig. 7.2 Maximum flow times for three flow patterns

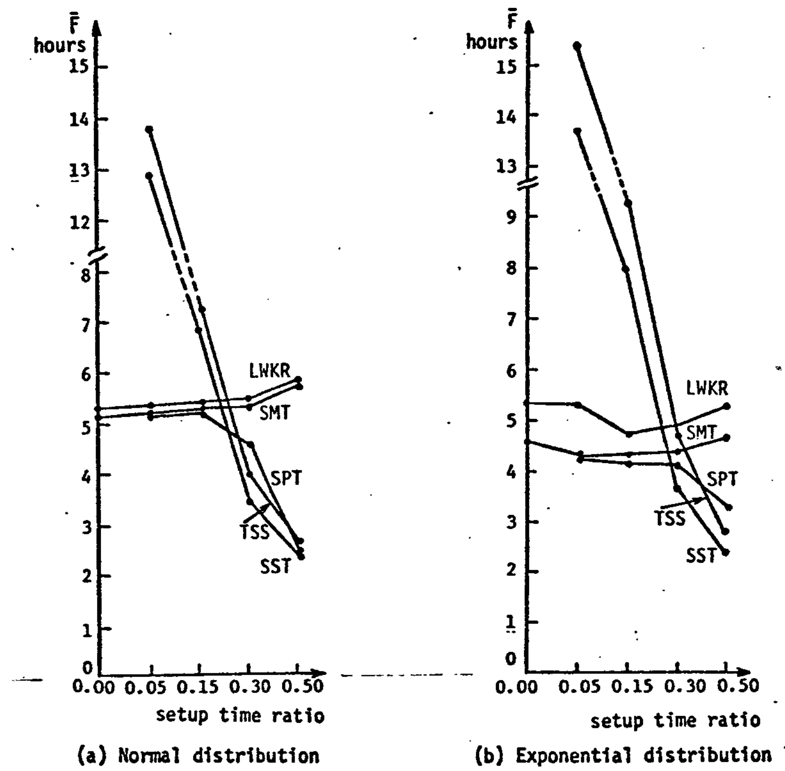


Fig. 7.3 Mean flow times for the production times generated from two distributions

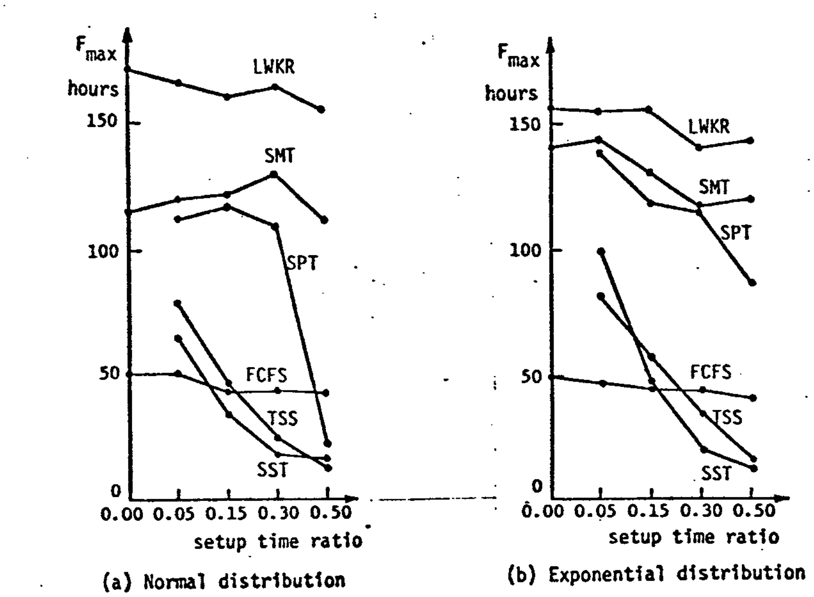


Fig. 7.4 Maximum flow times for the production times generated from two distributions

by the variation of setup times rather than that of processing times. When the standard deviation of the production times is set double, the value of the maximum flow time for the setup time ratio of 0.50 was over 100 hours, which was almost equal to the value for $R = 0.30$.

7.5 Conclusions

(1) The conventional scheduling rules, such as FCFS (First-Come, First-Served), SPT (Shortest Processing Time), and others, showed almost the same relative performances on the mean flow time for group scheduling.

(2) There was no significant difference in performances on the mean flow time and the maximum flow time for the different flow patterns for any scheduling rule except the FOPNR (Fewest Operations Remaining) and LWKR (Least Work Remaining) rules.

(3) The performances on the mean flow time and the maximum flow time for the well-known SPT and LWKR rules were also greatly influenced by the shop load for group scheduling. The ratio of the setup time to the processing time as well as the shop load influenced the performances for the SST (Shortest Setup Time) and TSS (Traveling Salesman Sequence) rules; however, the ratio of the range of the setup time to the mean setup time showed no influence on the performances for the four scheduling rules, SPT, LWKR, SST, and TSS.

(4) In group scheduling, where the relative size of the setup time to the processing time is large, the use of the scheduling rules, such as SST and TSS including the setup time and the setup group, seems to be desirable.

(5) The difference in production-time distributions is not critical in comparing scheduling rules.

CHAPTER 8 SUMMARY

Group technology, which is a technique to improve the productivity in small and medium lot size production, has gained major attention in manufacturing industries. In this thesis, group scheduling models based on the concept of group technology were developed in an attempt to achieve the benefits of group technology applications.

In the models under static conditions, the fundamental assumptions that jobs to be processed are classified into several groups and jobs within the same group are processed in succession, were made. Scheduling models of this new type required the determination of both the sequence of groups classified and the sequence of jobs in each group simultaneously.

The following conclusions were reached for the static group scheduling models.

(1) The single-stage group scheduling model was developed under three kinds of criteria —— the minimum mean flow time, the minimum weighted mean flow time, and the minimum total tardiness. For the minimum-mean-flow-time and the minimum-weighted-mean-flow-time problems, two theorems were given for optimally determining a group schedule (group and job sequences). Several theorems which specify the relative order of pairs of groups in an optimal group schedule were proved for the problem of minimizing the total tardiness. With the use of those theorems, efficient algorithms for determining the optimal and the near optimal group schedules were developed. The effectivenesses of the algorithms were verified with numerical examples.

(2) The single-stage group scheduling model with sequence-dependent setup times was developed under three kinds of criteria —— the minimum total elapsed time, the minimum mean flow time, and the minimum total tardiness. The problem with the objective of minimizing the total elapsed time was

shown to be reduced to the traveling salesman problem. In order to solve the minimum-mean-flow-time problem, the dynamic programming approach and the branch-and-bound method were applied. For determining a group schedule minimizing the total tardiness, the branch-and-bound algorithm was developed and a numerical example was shown.

(3) Theoretical analyses were made for the two-stage flow-shop scheduling problems with setup times separated and time lags when the objective was to minimize the total elapsed time. Theorems, which were extensions of Johnson's, were developed to determine the optimal schedules. They were extended to the two-stage flow-shop group scheduling. Furthermore, the special multistage flow-shop group scheduling problems, where there exist some well-defined relationships among the group setup times and the job processing times, were theoretically treated and a theorem and algorithms were given to determine the optimal group schedule for each of the cases.

(4) In order to solve the multistage flow-shop group scheduling problems, the branch-and-bound method was applied. For the problem of minimizing the total elapsed time, several lower bounds were developed and the effectivenesses of these were examined with numerical experiments. The machine-based lower bound was verified to be more effective than others. In addition, in the case of minimizing the weighted mean flow time, the effect of randomness of the weighting factors on the effectiveness of the branch-and-bound algorithm was tested with numerical experiments. The results showed that the algorithm was effective when the variation of the weighting factors was large.

(5) A multistage group scheduling model with variable job processing times and costs was developed. In this scheduling model, job processing times were assumed to be variable, depending on machining speeds for jobs, and hence, decisions were to be made as to the scheduling of groups and jobs and

the determining of machining speeds. The minimum total elapsed time with the minimum number of tardy jobs was employed as a scheduling criterion, and the minimum total production cost was employed as a criterion for determining optimal machining speeds. The analysis of the model presented was made under these criteria, and the optimizing algorithms for determining the optimal group schedule and the optimal machining speeds were proposed. The effectivenesses of the algorithms were verified with numerical examples.

In the last part of this thesis, the group scheduling model under dynamic conditions where jobs arrive at random over time was developed. In this model, it was assumed that jobs were classified into several setup groups and group setup times were sequence-dependent. The following conclusion was obtained for this problem.

(6) A simulation model was constructed and the simulation experiments were run to investigate the effect of types of flow patterns —— job-shop, near-flow-shop, and flow-shop patterns on flow time performances. Results showed that there was no significant difference in performances for the different flow patterns. In addition, the effect of the setup time on the performances for several scheduling rules was also studied in the experiments. Results indicated that the setup time played a critical role in group scheduling in those cases where the relative length of the setup time to the processing time was large.

ACKNOWLEDGMENTS

The author would like to express his sincerest appreciation to Dr. K. Hitomi, Professor of Osaka University, for his valuable guidance and encouragement.

The author would also like to express his gratitude to Dr. S. Makinouchi, Dr. T. Nishida, and Dr. K. Tezuka, Professors of Osaka University, for their critical reading of the manuscript and helpful suggestions.

The author is grateful to Dr. N. Nakamura, Associate Professor of Hiroshima University and Dr. M. Nakazima, Assistant Professor of Kyoto University of Industrial Arts and Textile Fibers, for their helpful suggestions.

The author's many thanks should be extended to Messrs. S. Takakuwa, T. Hamada, and M. Yokoyama, and K. Okuda, Assistants and Technician of Osaka University, for their assistance in preparing the manuscript.

REFERENCES

Chapter 1.

(1) Mitrofanov, S. P., Scientific Principles of Group Technology, translated by Harris, E., Yorkshire, National Lending Library, 1966.

(2) Opitz, H., Werkstucksystematik und Teilefamilienfertigung, Verlag W. Girardet, 1963/65/67.

(3) Nihon Kikai Shinko Kyokai, A Guide Book for Introduction of Group Technology, 1973.

(4) Petrov, V. A., Flowline Group Production Planning, Business of Publications, London, 1966.

(5) Burbidge, J. L., Production Planning, Heineman, London, 1971.

(6) Dedich, W. L., A. Soyster, and I. Ham, "The Optimal Formulation of Production Group Flowline," Proceedings of the 2nd North American Metal Working Research Conference, 1974, pp. 420-532.

(7) PERA, "Work Loading and Scheduling in Group Technology Cell," Production Technology, Report 282, 1974, pp.1-12.

(8) Conway, K. R., W. L. Maxwell, and L. W. Miller, Theory of Scheduling, Addison-Wesley Publishing Company, 1967.

(9) Baker, K. R., Introduction to Sequencing and Scheduling, John Willey and Sons, New York, 1974.

(10) Kan, A. R., Machine Scheduling Problems, Martinus Nijhoff, The Hague, 1976.

(11) Smith, W. E., "Various Optimizers for Single Stage Production," Nav. Res. Log. Quart., Vol. 3, No. 1, 1956, pp. 51-66.

(12) Moore, J. M., "An n Job, One Machine Sequencing Algorithm for Minimizing the Number of Late Jobs," Management Science, Vol. 15, No. 1, 1968, pp. 102-109.

(13) Johnson, S. M., "Optimal Two- and Three-Stage Production Schedules

with Setup Times Included," Nav. Res. Log. Quart., Vol. 1, No. 1, 1954, pp. 61-68.

(14) Akers, S. B. Jr., "A Graphical Approach to Production Scheduling Problems," Operation Research, Vol. 4, No. 2, 1956, pp. 244-245.

(15) Gapp, W., D. S. Mankekar, and L. G. Mitten, "Sequencing Operations to Minimize In-Process Inventory Costs," Management Science, Vol. 11, No. 3, 1965, pp. 476-484.

(16) Kurisu, T., "Single Machine Sequencing under Precedence Constraints," Technology Reports of the Osaka University, Vol. 27, 1977, pp. 27-34.

(17) Kurisu, T., "Two-Machine Scheduling under Required Precedence among Jobs," J. Opns. Res. Soc. Japan, Vol. 19, No. 1, 1976, pp. 1-13.

(18) Kurisu, T., "Two-Machine Scheduling under Arbitrary Precedence Constraints," J. Opns. Res. Soc. Japan, Vol. 20, No. 2, 1977, pp 113-131.

(19) Nakamura, N. and K. Hitomi, "Analysis of Production Scheduling (Optimization of Group Scheduling on Multiple Production System)," Trans. Japan Soc. Mech. Eng. Vol. 22, No. 361, 1976, pp. 2964-2975.

(20) Hitomi, K. and I. Ham, "Operation Scheduling for Group Technology Applications," CIRP Annals, Vol. 25, Hallway Ltd., Bern, Switzerland, 1976, pp. 419-422.

Chapter 2.

(1) Smith, W. E., "Various Optimizers for Single Stage Production," Nav. Res. Log. Quart., Vol. 3, No. 1, 1956, pp. 51-66.

(2) Moore, J. M., "An n Job, One Machine Sequencing Algorithm for Minimizing the Number of Late Jobs," Management Science, Vol. 15, No. 1, pp. 102-109.

(3) Conway, R. W., W. L. Maxwell, and L. W. Miller, Theory of Scheduling, Addison-Wesley Publishing Company, 1967.

(4) Baker, K. R., Introduction to Sequencing and Scheduling, John Wiley and Sons, New York, 1974.

(5) Yoshida, T., N. Nakamura, and K. Hitomi, "A Study of Production Scheduling (Optimization of Group Scheduling on a Single Production Stage)," Trans. Japan Soc. Mech. Eng., Vol. 39, No. 322, 1973, pp. 1993-2003.

(6) Yoshida, T., N. Nakamura, and K. Hitomi, "Group Production Scheduling for Minimum Total Tardiness-Part (I)," AIIE Trans., Vol. 10, No. 2, 1978, pp. 157-162.

(7) Wilkerson, L. J., and J. D. Irwin, "An Improved Algorithm for Scheduling Independent Tasks," AIIE Trans., Vol. 3, No. 3, 1971, pp. 239-245.

(8) Emmons, H., "One-Machine Sequencing to Minimize to Certain Functions of Job Tardiness," Operations Research, Vol. 17, No. 4, 1969, pp. 701-715.

Chapter 3

(1) Bellman, R., "Dynamic Programming Treatment of the Traveling Salesman Problem," J. Assn. for Computing Machinery, Vol. 9, No. 1, 1962, pp. 61-63.

(2) Held, M. and R. M. Karp, "A Dynamic Programming Approach to Sequencing Problems," J. Soc. Ind. and Appl. Math., Vol. 10, No. 2, 1962, pp. 196-210.

(3) Little, J. D. C., K. G. Murty, D. W. Sweeny, and C. Karel, "An Algorithm for the Traveling Salesman Problem," Operations Research, Vol. 11, No. 6, 1963, pp. 972-989.

(4) Gavett, J. W., "Three Heuristic Rules for Sequencing Jobs to a Single Production Facility," Management Science, Vol. 11, No. 8, 1965, 166-176.

(5) Karg, R. and G. L. Thompson, "A Heuristic Approach to Solving Traveling Salesman Problems," Management Science, Vol. 10, No. 2, 1964.

(6) Yoshida, T. and K. Hitomi, "A Study of Production Scheduling (Optimization of Group Scheduling with Sequence-Dependent Setup Times)," Trans. Japan, Soc. Mech. Eng., Vol. 43, No. 368, 1977, pp. 1531-1538.

(7) Yoshida, T., N. Nakamura, and K. Hitomi, "A Study of Production Scheduling (Optimization of Group Scheduling on a Single Production Stage)," Trans. Japan, Soc. Mech. Eng., Vol. 39, No. 322, 1973, pp. 1993-2003.

(8) Bellman, R., Dynamic Programming, Prince University Press, 1957.

(9) Lawler, E. L. and D. E. Wood, "Branch-and-Bound Method: A Survey," Operations Research, Vol. 14, No. 4, 1966, pp. 699-719.

(10) Baker, K. R., Introduction to Sequencing and Scheduling, John Wiley and Sons, New York, 1974.

(11) Nakamura, N. and K. Hitomi, "A Study of Production Scheduling (Optimization of Group Scheduling on the Multiple Production System)," Trans. Japan, Soc. Mech. Eng., Vol. 42, No. 361, 1976, pp. 2964-2975.

Chapter 4

(1) Johnson, S. M., "Optimal Two- and Three-Stage Production Schedules with Setup Times Included," Nav. Res. Log. Quart., Vol. 1, No. 1, 1954, pp. 61-68.

(2) Nabeshima, I., "The Order of n Items Processed on m Machines," J. Opns. Res. Soc. Japan, Vol. 3, No. 4, 1961, pp. 170-175.

(3) Nabeshima, I., "Some Extensions of the m-Machine Scheduling Problem," J. Opns. Res. Soc. Japan, Vol. 10, Nos. 1, and 2, 1967,

pp. 1-17.

(4) Smith, M. L., S. S. Panwalkar, and R. A. Dudek, "Flowshop Sequencing Problem with Ordered Processing Time Matrix," Management Science, Vol. 21, No. 5, 1975, pp. 544-549.

(5) Gupta, J. N. D., "Optimal Schedules for Special Structure Flowshops," Nav. Res. Log. Quart., Vol. 22, No. 2, 1975, pp. 255-269.

(6) Szwarc, W., "Special Cases of the Flow-shop Problem," Nav. Res. Log. Quart., Vol. 24, No. 3, 1977, pp. 483-492.

(7) Mitten, L. G., "Sequencing n Jobs on Two Machines with Arbitrary Time Lags," Management Science, Vol. 5, No. 3, 1959, pp. 293-298.

(8) Johnson, S. M., "Discussion: Sequencing n Jobs on Two Machines with Arbitrary Time Lags," Management Science, Vol. 5, No. 3, 1959, pp. 299-303.

(9) Nabeshima, I., "Sequencing on Two Machines with Start Lag and Stop Lag," J. Oper. Res. Soc. Japan, Vol. 5, No. 3, 1963, pp. 97-101.

(10) Yoshida, T. and K. Hitomi, "Multistage Production Scheduling with Setup-Time Consideration," Trans. Japan, Soc. Mech. Eng., Vol. 45, No. 395, 1979.

(11) Yoshida, T., "A Study of Production Scheduling (Theory of Multistage Group Scheduling to Minimize Total Elapsed Time)," Journal of Kobe University of Commerce, Vol. 30, Nos. 5 and 6, 1979, pp.57-76.

(12) Yoshida, T. and K. Hitomi, "Optimal Group Scheduling on Multiple Production Stages—Theoretical Analysis," Technology Reports of the Osaka University, Vol. 28, No. 1432, 1978, pp. 261-270.

(13) Yoshida, T., "Optimal Two-Machine Scheduling," Journal of Kobe

University of Commerce, fiftieth anniversary issue, April, 1979, pp. 253-266.

(14) Conway, R. W., W. L. Maxwell and L. W. Miller, Theory of Scheduling, Addison-Wesley Publishing Company, 1967.

(15) Kurisu, T., "Two-Machine Scheduling under Required Precedence among Jobs," J. Opns. Res. Soc. Japan, Vol. 19, No. 2, 1976, pp. 1-13.

(16) Kurisu, T., "Two-Machine Scheduling under Arbitrary Precedence Constraints," J. Opns. Res. Soc. Japan, Vol. 20, No. 2, 1977, pp. 113-131.

Chapter 5

(1) Baker, K. R., Introduction to Sequencing and Scheduling, John Wiley & Sons, Inc., 1974.

(2) Ignall, E. and L. E. Schrage, "Application of the Branch-and-Bound Technique to Some Flow-shop Scheduling Problems," Operations Research, Vol. 13, No. 3, 1965, pp. 401-412.

(3) Lomnicki, Z., "A Branch-and-Bound Algorithm for the Exact Solution of Three-Machine Scheduling Problem," Operational Research Quarterly, Vol. 16, No. 1, 1965, pp. 89-101.

(4) Brown, A. and Z. Lomnicki, "Some Applications of the Branch-and-Bound Algorithm to the Machine Scheduling Problem," Operations Research Quarterly, Vol. 17, No. 2, 1966, pp. 173-186.

(5) McMahon, G. B. and P. Q. Burton, "Flowshop Scheduling with the Branch-and-Bound Method," Operations Research, Vol. 15, No. 3, 1967, pp. 473-481.

(6) Nakamura, N. and K. Hitomi, "A Study of Production Scheduling (Optimization of Group Scheduling on the Multiple Production System)," Trans. Japan, Soc. Mech. Eng., Vol. 42, No. 361, 1976, pp. 2694-2975

(7) Yoshida, T., Nakamura, N. and Hitomi, K., "A Study of Group Scheduling," Transactions of Japan Industrial Management Association, Vol. 28, No. 3, 1977, pp. 323-328.

Chapter 6

(1) Hitomi, K., "Analysis of Production Model (Optimal Scheduling and Machining Speed Decision of Multiproduct, Multistage Production System)," Trans. Japan Soc. Mech. Eng., Vol. 43, No. 367, 1977, pp. 1142-1147.

(2) Hitomi, K. and I. Ham, "Group Scheduling Technique for Multiproduct, Multistage Manufacturing System," Journal of Engineering for Industry, Trans. ASME, Vol. 99, Series B, No. 3, 1977, pp. 759-765.

(3) Ham, I. and K. Hitomi, N. Nakamura, and T. Yoshida, "Optimal Group Schedule and Optimal Machining Speeds of Multiproduct, Multistage Machining System for Group Technology," Trans. Japan Soc. Prec. Eng., Vol. 42, No. 6, 1976, pp. 14-19.

(4) Conway, R. W., W. L. Maxwell, and L. W. Miller, Theory of Scheduling, Addison-Wesley Publishing Company, 1967.

(5) Baker, K. R., Introduction to Sequencing and Scheduling, John Wiley & Sons, Inc., 1974.

(6) Ham, I., K. Hitomi, N. Nakamura, and T. Toshida, "Optimal Group Scheduling and Machining-Speed Decision under Due-Date Constraints," ASME Paper, No. 78-WA/PROD-39, 1978, pp. 1-7.

(7) Yoshida, T. and K. Hitomi, "Optimal Group Scheduling for Minimum Total Elapsed Time with Number of Tardy Jobs," Preprint, Japan Soc. Mech. Eng., No. 783, 1978, pp. 49-52.

(8) Hitomi, K., "Economical and Optimal-Seeking Machining," Metal Processing Machine Tools, Instituto per le Ricerche di

Tecnologia Meccanica, Vico. Canavese, Italy, 1970, pp. 137-153.

(9) Moore, J. M., "An n Job, One Machine Sequencing Algorithm for Minimizing the Number of Late Jobs," Management Science, Vol. 15, No. 1, 1968, pp. 102-109.

(10) Hitomi, K., Theory of Planning for Production, Yuhikaku, Ltd., Tokyo, 1975.

(11) Hitomi, K., Manufacturing Systems Engineering, Taylor & Fransis Ltd., London, 1978.

Chapter 7

(1) Baker, C. T., and B. P. Dzielinski, "Simulation of Simplified Job Shop," Management Science, Vol. 6, No. 3, 1960, pp. 311-323.

(2) Conway, R. W., "Priority Dispatching and Work-in-Process in a Job Shop," J. Ind. Eng., Vol. 16, No. 2, 1965, pp. 123-130.

(3) Conway, R. W., "Priority Dispatching and Job Lateness in a Job Shop," J. Ind. Eng., Vol. 16, No. 4, 1965, pp. 228-237.

(4) Conway, R. W., W. L. Maxwell, and L. W. Miller, Theory of Scheduling, Addison-Wesley Publishing Company, 1967.

(5) Baker, K. R., Introduction to Sequencing and Scheduling, John Wiley & Sons Inc., 1974.

(6) Panwalker, S. S. and W. Iskander, "A Survey of Scheduling Rules," Operations Research, Vol. 25, No. 1, 1977, pp. 45-61.

(7) Hottenstein, M. P., "Expediting in Job-Order-Control Systems; A Simulation Study," Trans. AIIE, Vol. 2, No. 1, 1970, pp. 46-54.

(8) Baker, K. R., "Priority Dispatching in the Single Channel Queue with Sequence-Dependent Setups," J. Ind. Eng., Vol. 19, No. 4, 1968, pp. 203-206.

(9) Wilbrecht, J. R. and W. B. Prescot, "The Influence of Setup Time on Job Shop Performance," Management Science, Vol. 16,

Series B, 1969, pp. 274-289.

(10) Hitomi, K., N. Nakamura, T. Yoshida, and K. Okuda, "An Experimental Investigation of Group Production Scheduling," Proceedings of 4th ICPR, Preprint 5.12, 1977. pp. 1-11.

(11) Hitomi, K., N. Nakamura, T. Yoshida, and K. Okuda, "A Study of Production Scheduling (A Simulation Analysis of Group Scheduling)," Preprint, Japan Soc. Mech. Eng., No. 774, 1977, pp. 104-106.