Dynamic Optimization of Job Allocation Using Constant Job-Mix Stages and Priority Factors

by

Paul Massicotte

Department of Mechanical Engineering
McGill University
Montreal, Canada

May 1997

A thesis submitted to the faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of Master of Engineering

0-612-37271-5

ABSTRACT

The dynamic optimization proposed in this work uses a linear programming technique to maximize the quantity of job orders processed on the machines at each constant job-mix stage. Priority factors guide the sequential allocation of partial and complete setups by ranking job/machine combinations in order of processing performance and capacity to meet due dates.

The job allocation is governed by an algorithm which constitutes the backbone of the dispatching software that was developed and used to solve the examples presented in this thesis.

Priority factors offer an effective mean of guiding the selection of setups by ranking job/machine combinations by processing speed and capacity to meet due times. The necessity for priority factors becomes more important as setup times increase in relation with quantities to be manufactured and the processing times. Priority factors also guide the allocation of partial setups on the machines to speed up job completion.

The job allocation system has a substantially potential for providing shorter makespan than the Shortest Operating Time methodology by increasing machine utilization.

Résumé

La technique d'ordonnancement présentée dans ce rapport fait appel à la programmation linéaire pour maximiser les quantités executées sur des machine à chaque stage de mixte de travail invariable. Des facteurs de priorité guident l'allocation séquentielle d'installations préparatoires partielles ou totales en ordonnant les combinaisons de travail/machine par ordre de performance et de capacité à rencontrer les dates prévues de production.

Le logiciel d'ordonnancement qui a permis d'obtenir les résultats aux exemples contenus dans cet ouvrage fut développé à partir d'un algorithme élaboré au cours de cette étude.

Le système d'allocation des tâches proposé permet de compléter les bons de travail plus rapidement que la méthode d'allocation selon les plus courts temps d'opération. Plus les temps préparatoires sont importants par rapport aux temps d'opération et des quantités en cours, plus il devient important d'utiliser de facteurs de priorité. La préparation partielle des tâches permet de raccourcir les temps de production en accélérant le début des opérations.

Statement of Originality and Contribution to Knowledge

The author of this thesis claims originality for the development of the following concepts:

- The allocation of jobs using time based and processing speed based priority factors within constant job-mix stages.
- Partial setups.
- The sequential allocation of jobs within stages.

The sequential allocation of jobs using priority factors and partial setups maximizes the quantity produced at each constant job-mix stage by taking advantage of linear programming and dynamic optimization.

Acknowledgements

I would like to extend my sincere gratitude to everybody who helped me complete this study.

The author is especially grateful to Professor Lucian Kops whose thoughtful advice and commitment to simplicity will remain a source of motivation. Thanks to my friend Louis Brunet for his constructive suggestions and dedicated proofreading.

My special thanks to my mother, father and my partner Mika, whose love, support and encouragement will always sustain me.

Contents

List of Figures

List of Tables

List of Notations

c_{Tji}	Time based priority factor for job order j at stage i
c_{Pjmi}	Processing speed based priority factor for job order j on machine m at stage i
C_{PTjmi}	Processing speed and time based priority factor of job order j on machine m at stage i
i	Stages, i = 1, 2, ...
j	Job order index, j = 1, 2, ...
j_D	Jobs being dispatched
j_{rk}	Job having its setup revised at iteration k
j_{Sm}	Job setup on machine m
k	Iteration, k = 1, 2, ...
m	Machines, m = 1, 2, ..., M
M	Total number of machines
Q_j	Quantity specified in job order j
r_{Ami}	Availabiltiy of machine m at stage i (1 if available, 0 otherwise)
r_{Cjmi}	Capacity to process job order j on machine m at stage i (1 if able, 0 otherwise)
t_{Aj}	Time job order j becomes available
t_{Ami}	Time available for processing jobs on machine m at stage i
t_{Dj}	Due time of job order j
t_{Pjm}	Processing time of job j on machine m

t_{SEi} Time stage i ends

t_{Sjm} Setup time of job j on machine m

t_{SRjm} Time remaining to complete setup of job j on machine m

t_{SSi} Time stage i starts

t_{UEm} Time at which machine m becomes available, following maintenance

t_{USm} Time machine m becomes unavailable, due to scheduled maintenance

x_{jmi} Quantity produced of job order j on machine m at stage i

x_{jmki} Quantity proposed of job order j on machine m at iteration k of stage i

Chapter 1

Introduction

Job allocation on machine tools is one of the most important and difficult production scheduling activity affecting factory performance. Job allocation has implications for the utilization of machine tools, shop efficiency, work-in-process and profitability of the operations. The allocation is a complex process as many parameters such as large variety of jobs, operations, suitability and availability of machine tools, setups, imposed release and completion dates, must be taken into account.

The dynamic optimization proposed in this work uses linear programming to maximize the quantity of job orders processed on the machines at each constant job-mix stage. Priority factors guide the sequential allocation of partial and complete setups by ranking job/machine combinations in order of processing speed and capacity to meet due dates.

The development of a dispatching software based on the work on constant job-mix stages by Natarajan[1] inspired the elaboration of original concepts. Long hours of C language [2] programming were rewarded when the author finally discovered, by imposing the principle of sequential allocation of setups and quantities, one avoids the endless cycles of setup and machine swapping which occur otherwise. Priority factors were added to enable the algorithm to allocate jobs to the most appropriate job/machine combinations when setup times were large compared to the total processing time of job orders. Partial setups further reduce the makespan – the total time required to completely process all jobs.

The following sections are organized to inform the reader on job shop scheduling and to demonstrate the utilization and benefits of the job allocation system developed during this investigation.

Chapter 2 presents a critical review of the literature on job shop scheduling. Numerous dispatching heuristics and combinatorial optimization methods influencing the job allocation on machines are reviewed and criticized.

Chapter 3 contains an explanation of the concepts involved in the job allocation system. Jobs are categorized with respect to their status which is updated by the discretization of stages and the dynamic optimization. The computation of priority factors precedes the linear programming formulation, the definition of partial setups and details on the mechanism involved in the sequential allocation.

Chapter 4 elaborates on the data required by the allocation system and presents an outline of the algorithm. The modules necessary to form stages, to compute priority factors, to formulate and solve the linear programming equations, to complete the sequential allocation are expanded to 15 operational steps.

The algorithm is applied to a problem in Chapter 5. The computation required at each step is explained and detailed results are tabulated.

Chapter 6 presents examples of parameters which influence dispatching. Detailed independent examples are run on the job allocation software to demonstrate the handling of setup times, the effect of quantities on priority factors, the enforcement of due dates, partial setups and the sequential allocation.

Chapter 7 compares the results obtained in the example of Chapter 5 to the ones compiled from Shortest Operating Time (SOT) dispatching rule. The benefits of the new approach become apparent by analyzing the machine utilization and hastening of completion time.

The conclusions in Chapter 8 are followed by recommendations for further study.

Chapter 2

Literature Review

The literature proposes numerous heuristics to allocate n jobs on m machines optimally or quasi-optimally. Some methods offer pragmatic solutions while others concentrate on the theoretical aspect of job allocation. Garey and al. [3] corroborated the computational complexity of the job shop scheduling problem by demonstrating its NP hard nature. Many algorithms propose to minimize the sum of the completion times, total flow time, makespan, waiting time or Work In Process (WIP). Others try to maximize the overall machine tool utilization.

The allocation of jobs is often performed by simple dispatching rules. The most frequently referred ones include Shortest Processing Time (SPT), Shortest Finish Time (SFT), Earliest Release Time (ERT), Earliest Finish Time with Alternative operations considered (EFTA), First-In-First-Out (FIFO), Last-In-First-Out (LIFO).

The solutions proposed by academia has progressed considerably. Muth an Thompson

[4] created a precedent by publishing their 10x10 job-shop problem. It focused the attention to a much more complex issue than the single machine system solved by Johnson's rule [5]. Powell [6] used the simplex method to minimize the total operations cost by allocating scarce resources to alternative jobs. Linear Programming (LP) made it feasible to consider all the possible alternatives but the influence of setup times on variable size batches was neglected. Nasr and Elsayed [7] decomposed the job-shop scheduling problems in sub-problems and minimized the completion time by mixed integer programming. The fact that jobs had to be available at time zero limited the application of this approach. Kops and Natarajan [8] handled the unsynchronised release and completion time of jobs by introducing a constant job-mix partitioning scheme based on the scheduled flow of jobs. Linear programming was used to maximize the quantity produced at each stage. Kops and Natarajan [9] followed up on their recommendation in there previous paper on constant job-mix stages and incorporated setup time considerations. Their work deals with setup times which are small compared to the total processing time of the jobs.

Fuzzy logic has also been applied to job allocation. Balazinski and Kops [10] took advantage of the fundamentals of fuzzy logic [11] to allocate jobs to the most suitable machine. Bugnon and al. applied fuzzy logic to real time control of task allocation [12]. The difficulties encountered in the search of an appropriate correlation severly limits the more general and widespread application of fuzzy logic to job allocation problems.

Reeves [13] mentioned that a fairly small case involving about 50 jobs solved using a Branch-and-Bound (B&B) methodology takes so long to solve that it is not practical. Hybrid methods using B&B and Mixed Integer Linear Programming (MILP) consume even more time. McMahon and Florian [14] proposed a B&B method that minimize the maximum lateness of jobs subject to ready times and due dates. The methodology proposed a solution at every node to a maximum of 500 nodes. Carlier [15] applied a B&B method to conjunctive graphs and minimized the makespan of a one-machine sequencing problem for up to 10 000 jobs. Carlier and Pinson [16] expanded the one-machine problem to the job-shop problem by optimizing the complexity of local algorithms. Schedules nearing the optimum were obtained for 50 jobs on 10 machines. The adaptive branching rules for B&B proposed by Potts [17] reduced computation time significantly. However multi-machine scheduling remained problematic. Balas tackled machine sequencing by applying the B&B approach to find the mini-maximal path of disjunctive graphs [18].

Hybrid systems capable of integrating search procedures and dispatching rules show promising results. Search procedures are implemented for effectiveness and appropriate dispatching rules are selected according to their corresponding efficiency. With increasing computing power, the effective iterative improvement may become an excellent supplement to efficient one-pass heuristics.

Post-processing of the initial schedule can be performed by algorithms which engender techniques such as the shifting bottleneck, Tabu Search (TS) and Genetic algorithm (GA). Adams, Balas and Zawack introduced the shifting bottleneck [19] procedure that performed local reoptimization by repeatedly solving certain one-machine scheduling problems. Ivens and Labrecht extended the shifting bottleneck procedure to real-life applications[20]. They proposed some improvements to optimize the one-machine sub-problems.

Glover established Tabu Search (TS) as a strategy for combinatorial manipulations by publishing its fundamentals [21] and confirmed its importance by the publication of a user's guide on the topic [22]. TS compounds flexible memory structures, strategic restrictions and aspiration levels. A tabu list is a "what-not-to-do" list. Laguna and Velarde demonstrated the relevance of TS by demonstrating its relevance in just-in-time scheduling of parallel machines [23], The meta-heuristic local search iterative improvement approach developed by Dake and Batta reduced the makespan of the n jobs m machines job shop scheduling problem through Active Chain Manipulation [24]. The proposed algorithm balanced the efficiency of dispatching rules with general effectiveness of the solution provided by TS.

Job shop scheduling should also benefit from the capacity of genetic algorithms to generate and consider an extremely large sample of possible allocations[25].

The concepts unveiled in the scheduling of a single machine with controllable processing times and compression costs [26] and the continuous flow models of manufacturing systems [27] may influence the market approach to holonic manufacturing[28].

Neophytes to the field of job shop scheduling could greatly benefit by reading on the mathematical implications of job allocation. The article published by Bjorndal and al [29] summarizes the trends in combinatorial optimization. The study published by Lourenco [30] elaborated on the strengths, weaknesses and limits of local search and large-step optimization methods influencing dispatching The review of job shop scheduling techniques completed by Blazewicz et al. [31] and the book on modern heuristic techniques by Reeves [32] presented an exhaustive overview of the field.

Chapter 3

Optimization using Constant Job Mix Stages, Priority Factors, Partial Setups and Sequential Allocation

This chapter explains the general concepts used by the job allocation algorithm. The discretization of stages dictates job status and leads to dynamic optimization. Jobs to be produced in a stage are dispatched to machines through a combination of manipulations involving priority factors, Linear Programming, sequential allocation heuristic and partial setups. Computational requirements are also outlined.

3.1 Job Status

As shown on Figure 3.1, every job occupies a succession of three distinct status in the allocation process. A job order waits until the Dispacher selects it to join a stage. Once a job is completely assigned to a single or a number of machines it is considered allocated.

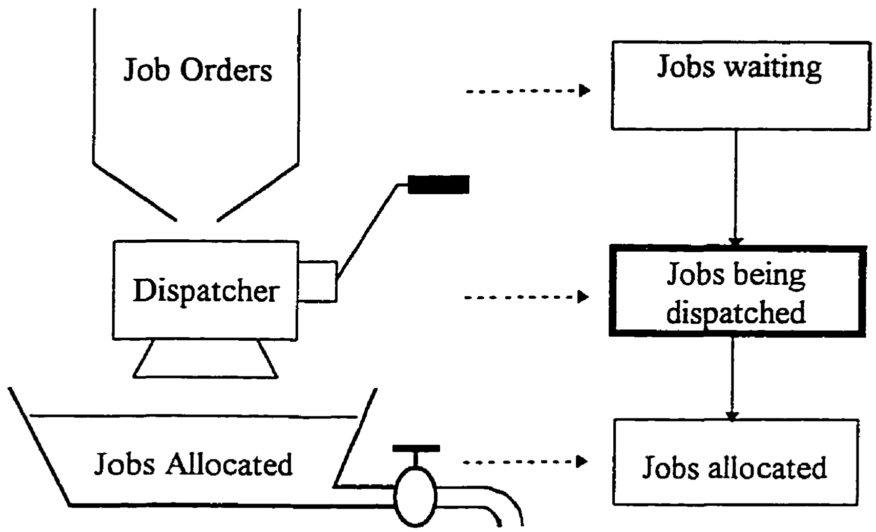

Fig 3.1 Job status

3.2 Discretizing Stages

Jobs being dispatched belong to a constant job mix (CJM) stage. Natarajan and Kops[33] defined CJM stages as the time period encompassing a fixed number of jobs. Stages begin or end with the arrival or completion of jobs. Figure 3.2 graphically represents the discretization of stages.

In the example illustrated in Figure 3.2, stage 1 comprises job A only. It extends from the arrival of job A (t_{AA}) to the arrival of job B (t_{AB}). Stage 2 starts with the two preceding jobs and ends with the arrival of job C (t_{AC}). Stage 3 has 3 jobs and terminates at the due date of job B (t_{DB}). Stage 4 ends with the release of job C (t_{DC}). Stage 5 contains job A only. The arrival of job D (t_{AD}) causes the beginning of Stage 6. Stage 6 finishes with job A leaving (t_{DA}). The final stage of this example contains only job D.

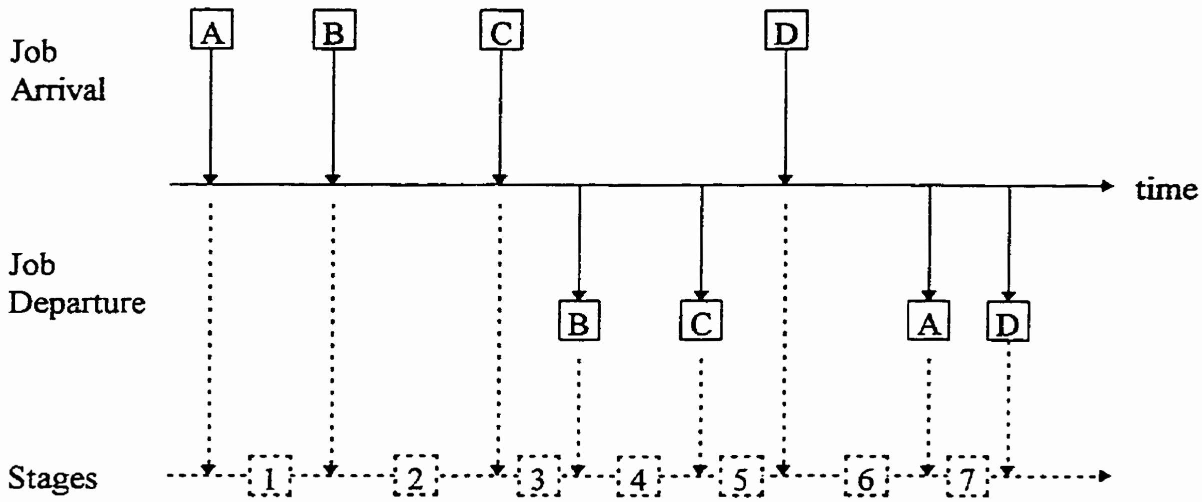

Fig 3.2 Discretization of stages

The boundaries of a stage are independent of the total number of jobs being dispatched. A stage can embody a single as well as an extremely large number of jobs. A stage could also contain no job. In such a case, all previous jobs would have been due or completed by the end of the preceding stage. And, there would be a delay before another job becomes available.

Stages arise according to the available (t_{Aj}) and due dates (t_{Dj}) of jobs. A job starting after its time available (t_{Aj}) or completed before its due date (t_{Dj}) will influence neither the actual stage boundaries nor subsequent stages boundaries.

3.3 Dynamic Optimization

The discretization into stages transforms the complex job allocation problem into a series of simpler problems that can be optimized dynamically [34]. The CJM stages are linked to each other such that the resulting state of a stage becomes the input to the subsequent stage.

The allocation performed by the Dispatcher allocates jobs on machines at each stage through a combination of heuristics and Linear Programming (LP). The previous section explained the discretization of stage. The priority factors explained in the next section will be incorporated in the LP equations of section 3.5. The distribution of jobs on machines proposed by the LP will be used by the sequential allocation presented in section 3.7 to finalize setups and quantities processed on machines.

The Dispatcher will allocate setups and quantities from the data shown on Figure 3.3. The discretization in CJM stages yielded the stage start and end time and jobs belonging to the list of jobs to be dispatched. The due date is specified in the job order and remains unchanged. LP and the sequential allocation will modify the quantities, setups and time remaining to complete. The start and end time of the unavailabilily of machines contained in the maintenance data enters calculations for the time available on each machine. The setup and processing time are selected from the tables on machine data. The Dispatcher will process these data to schedule setups and quantities to produce on the machines.

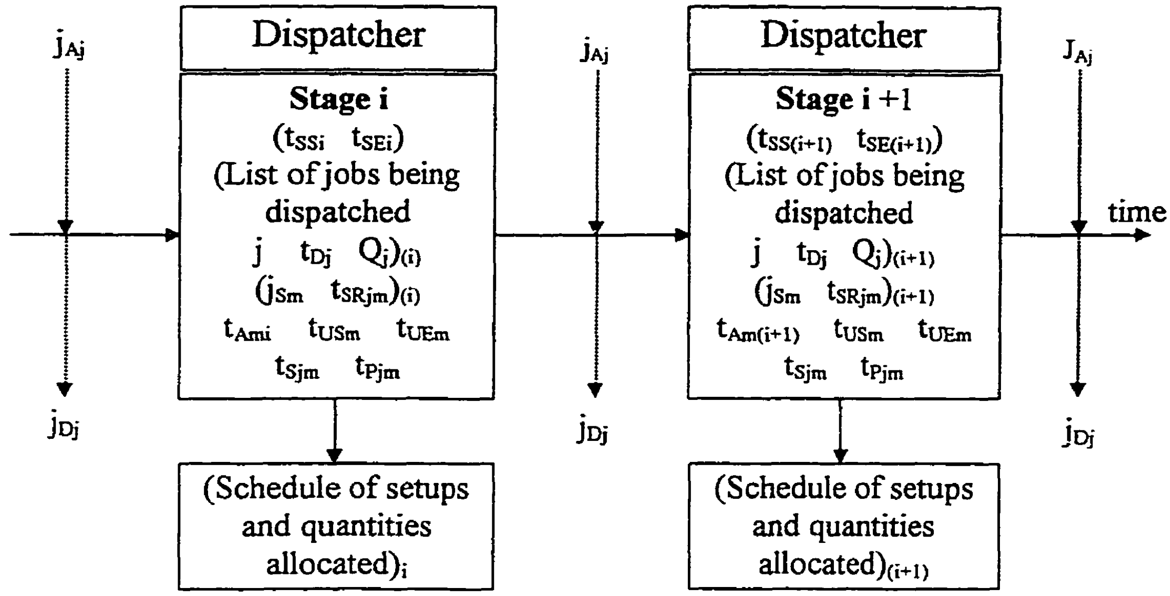

Fig 3.3 Dynamic allocation

It is advantageous to partition the scheduling horizon and solve sub-problems because the computational effort and time required to solve an LP is proportional to the size of the problem.

3.4 Priority Factors

The dynamic allocation of jobs on machines starts with the computation of priority factors. The processing speed and time based priority factor (C_{PTjmpi}) combines the influence of the time based priority factor (C_{Tji}) and the processing speed based priority factor (c_{Pjmi}).

The time based priority factor (c_{Tji}) enforces timely completion of jobs. The weight of the time based priority factor (c_{Tji}) increases as stages get closer to the due date (t_{Dj}).

$$c_{PTjmi} = c_{Tji} \times c_{Pjmi}$$

where $$c_{Tji} = \frac{t_{SEi} - t_{SSi}}{t_{Dj} - t_{SSi}}$$

and $$c_{Pjmi} = r_{Ami} r_{Cjm} \frac{1}{t_{Pjm} + \frac{t_{SRjm}}{Q_j}} \qquad \text{if } j_{Sm} \neq j \Rightarrow t_{SRjm} = t_{Sjm}$$

The processing speed based priority factor (c_{Pjmi}) embodies setup time considerations. The machine must be available (r_{Ami}) to process jobs in the stage under consideration. The machine must also have the capacity (r_{Cjmi}) to produce the job totally or partially.

It is important to notice that the sequential allocation process may modify the availability and the capacity of machines between iterations in a same stage. The influence of the remaining setup time (t_{SRjm}) is proportional to the quantity (Q_j) being allocated. If the existing setup (j_{Sm}) of a machine does not correspond to the job (j), the full setup time of the job (t_{Sjm}) on the machine is used in the computation. Once the machine setup is completed, neither of the remaining setup time (t_{SRjm}) nor the quantity (Q_j) influence this priority factor.

3.5 Linear Programming (LP) Formulation

The job distribution obtained from LP and the sequential allocation process described in section 3.7 are interdependent. The LP proposes the allotment of jobs on machines.

The objective function of the LP aims at maximizing the allocated quantities (x_{jmi}) of jobs being dispatched in the stage. The LP solver successively allocates quantities starting from the job-machine combination with the highest processing perfomance and time based priority factor (C_{PTjmi}). Jobs already allocated will not be part of the LP as their priority factor was set to zero through the sequential allocation heuristic.

The LP formulation is constrained by the quantities of each job (Q_j) to allocate and the time available on machines (t_{Ami}). Jobs can only be alloted to capable machine ($r_{Cjmi} = 1$) and available machines ($t_{Ami} = 1$). The availability and capacity of machines is constantly revised by the sequential allocation heuristic. The sum of quantities (x_{jmi}) alloted of a given job may be smaller than the total quantity to produce (Q_j) if the job is not due by the end of the stage; it must be equal to Q_j if it is due by the stage end. The total time required to process all quantities (x_{jmi}) allocated to a given machine in a stage cannot exceed the time available on the machine (t_{Ami}). The sequential allocation heuristic handles the setup time portion of the time available. It also performs a partial setup if the time does not allow for a complete setup. Of course, quantities (x_{jmi}) can never be negative. The LP equation is solved using any linear programming package.

Jobs are distributed on machines in accordance with the following generalized LP formulation:

Objective: $\text{MAX}\sum_{m=1}^{M}\sum^{j_D} c_{PTjmi}x_{jmi}$

Subject to: $\sum_{m=1}^{M} r_{Ami}r_{Cjm}x_{jmi}$ $\le Q_j$ or $= Q_j$ if last stage for this job — for all jobs being dispatched

$$\sum_{j=j_{r(k-1)}} r_{Ami}r_{Cjmi\ jmi}t_{Pjm} \le t_{Ami} \qquad \text{for } m = 1\ldots M$$

$$x_{jmi} \ge 0 \qquad \text{for } j_D$$

The LP formulation evolves through each iteration of a stage. The first iteration of a stage covers all the jobs being dispatched in the stage. Once a job has been distributed on machines, it does not have to be considered by the LP anymore for the stage under consideration. Therefore, the number of terms to optimize by the LP will decrease iteration by iteration. The computational time will also be reduced accordingly.

The quantities generated by LP are rounded off to the lower integer for practical reasons. Machines produce complete parts. A machine may be alloted a portion of the quantity to produce. But, a machine must complete each piece allocated of a given job.

Integer linear programming was not used because it is slow. It also has a tendency to produce sub-optimal local maxima and get caught in its branch and bound process.

The LP must assume that the setup times and processing times are constant. In reality, it depends on the operator and the machine. The setup time and processing time for labour intensive manipulations may decrease as the operator's skills improve. The times on automated machines may also be influenced by the reliability of the machine for the given operation. Unexpected breakdowns will unfavorably influence the performance.

3.6 Partial Setups

The sequential allocation heuristic performs a partial setup when the time available does not allow to complete the setup. In such a case, the time remaining for setup may be completed in subsequent stages if the dispatcher retains the same machine to process the job.

3.7 Sequential Allocation

The sequential allocation distributes the quantities proposed by LP and schedules setups on machines in accordance with the heuristic shown on Figure 3.4. LP and sequential allocation interact through common parameters such as the capacity factor (r_{Cjmi}), the machine availability (r_{Ami}) and time available on machines (t_{Ami}) at every iteration of a stage.

The sequential allocation requires two successive iterations to allocate setups and quantities of a job on the machines. Jobs are distributed one by one, iteration by iteration. starting with the job having the earliest due date and ending having the job with the latest due date. Except for the first and last iteration of a stage, quantities allocated for the job for which setups were performed in the previous iteration and the setups for the next job to be allocated are saved at each iteration.

A first iteration is necessary to perform a setup on the machines that were allocated quantities of the job under consideration. The capacity of machines that were not allocated any quantity of the job being distributed is disabled ($r_{Cjmi} = 0$) to eliminate the risk of having quantities of the same job allocated to them in subsequent iterations. The time available on a machine (t_{Ami}) will be decreased by the setup time that can be completed. No change will be necesary to the availability of a machine if the setup was completed in a previous stage. If a setup that was started in a previous stage can be completed, the time available will be decreased accordingly. If only a partial setup can be performed, there will be no time available on the machine and the machine will not have the capacity to process any other jobs in this stage. The setup performed are saved at each iteration.

The quantities allocated are saved in the following iteration. The processing time required is subtracted from the time available (t_{Ami}) on the machine at the stage. The capacity of the machine to process this job will be nullified ($r_{Cjmi} = 0$) as the same job cannot be allocated on the same machine twice in the same stage. Afterward, the next job to be distributed is selected and setups are updated accordingly.

The first and last iteration of a stage differ in the following way. Setups will be revised for the job with the earliest due date to be distributed but no quantities are allocated in the first iteration because no job had its setup revised previously. The last iteration will allocate quantities of the job with the latest due date and will not revise any setups because they will all have been considered previously. There will be a maximum of one iteration exceeding the total number of jobs being dispatched in a stage.

3.8 Computer Requirements

The examples presented in this work were solved by the Job Allocation Software that was developed during this master's degree to reproduce the computations of the Dispatcher. The computer code was written in C language and was compiled on a 486 PC compatible with 16 megabytes of RAM running at 66 MHz. The software can run on any PC 386 or higher. The sample text files used to input information concerning job orders, machine and maintenance data and results are shown in appendix A.

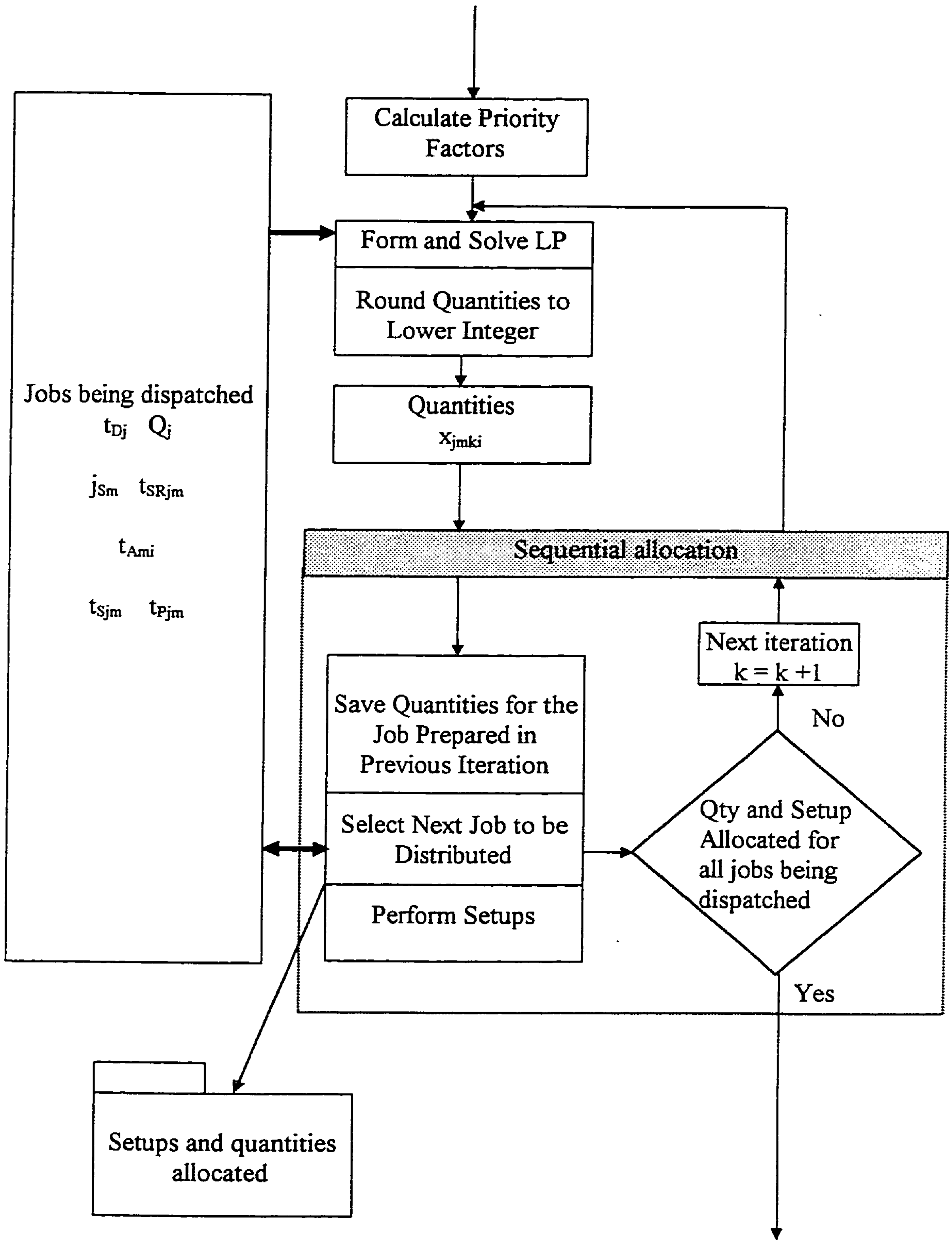

Fig 3.4 Sequential allocation heuristic

Chapter 4

The Job Allocation Algorithm

This chapter outlines the operations executed by the job allocation algorithm. A step by step explanation of the system follows the enumeration of the data required.

4.1 Data Required

The Dispatcher allocates jobs for maximizing quantities processed on the machines at each stage after accounting for machine availability, processing time and setup time. The job parameters required for the allocation are shown on Figure 4.1. They comprise an available date (t_{Aj}), a due date (t_{Dj}) and a quantity (Q_j) to be dispatched for each job. Machine setup (t_{Sjm}) and processing times (t_{Pjm}) are provided in a tabular form as shown in Figure 4.2. A processing time (t_{Pjm}) set to zero indicates that the machine cannot produce the given job. The system will not allocate jobs on machines undergoing maintenance. Maintenance periods are contained in a list as shown in Figure 4.3. The unavailable time of a machine m starts at t_{USm} and terminates at t_{UEm}.

JOB ORDER data

j	t_{Aj}	t_{Dj}	Q_j
1	t_{A1}	t_{D1}	Q_1
2	t_{A2}	t_{D2}	Q_2
...	.....	.....	....
...	.....	.....	....
J	t_{AJJ}	t_{DJ}	Q_J

Fig 4.1 Job order data

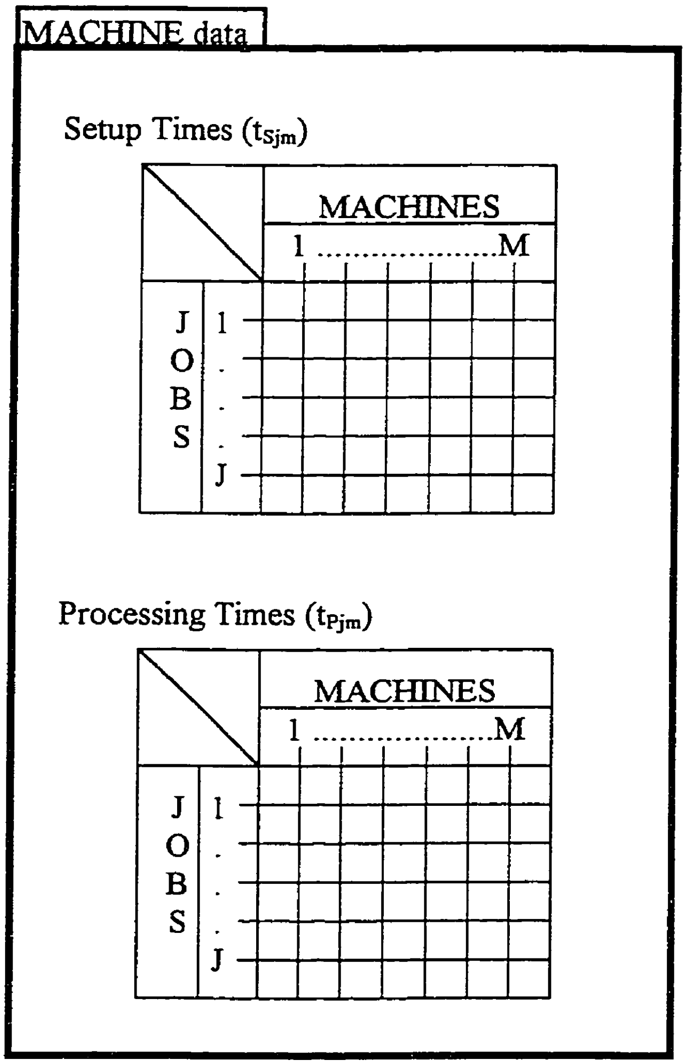

Fig 4.2 Machine setup and processing times

MAINTENANCE data		
m	t_{USm}	t_{UEm}
...	.....	.....
...	.....	.....
...	.....	.....
...	.....	.....
...	.....	.....
...	.....	.....
...	.....	.....

Fig 4.3 Machine maintenance data

4.2 Algorithm Procedure

The flow chart of the job allocation system illustrated on Figure 4.4 presents the modules controlled by the Dispatcher. The operations occuring in each module are explained with the help of block diagrams.

The job allocation system performs its operations in two loops. The primary loop governs the formation of stages, operations for the sequential allocation of jobs within each stage and stops when the last stage has been allocated . The secondary loop, commonly referred to as a nested loop, controls the sequential allocation of jobs within each stage. The total number of stages and the number of jobs being dispatched at each stage depends on the time and quantity specified in job orders.

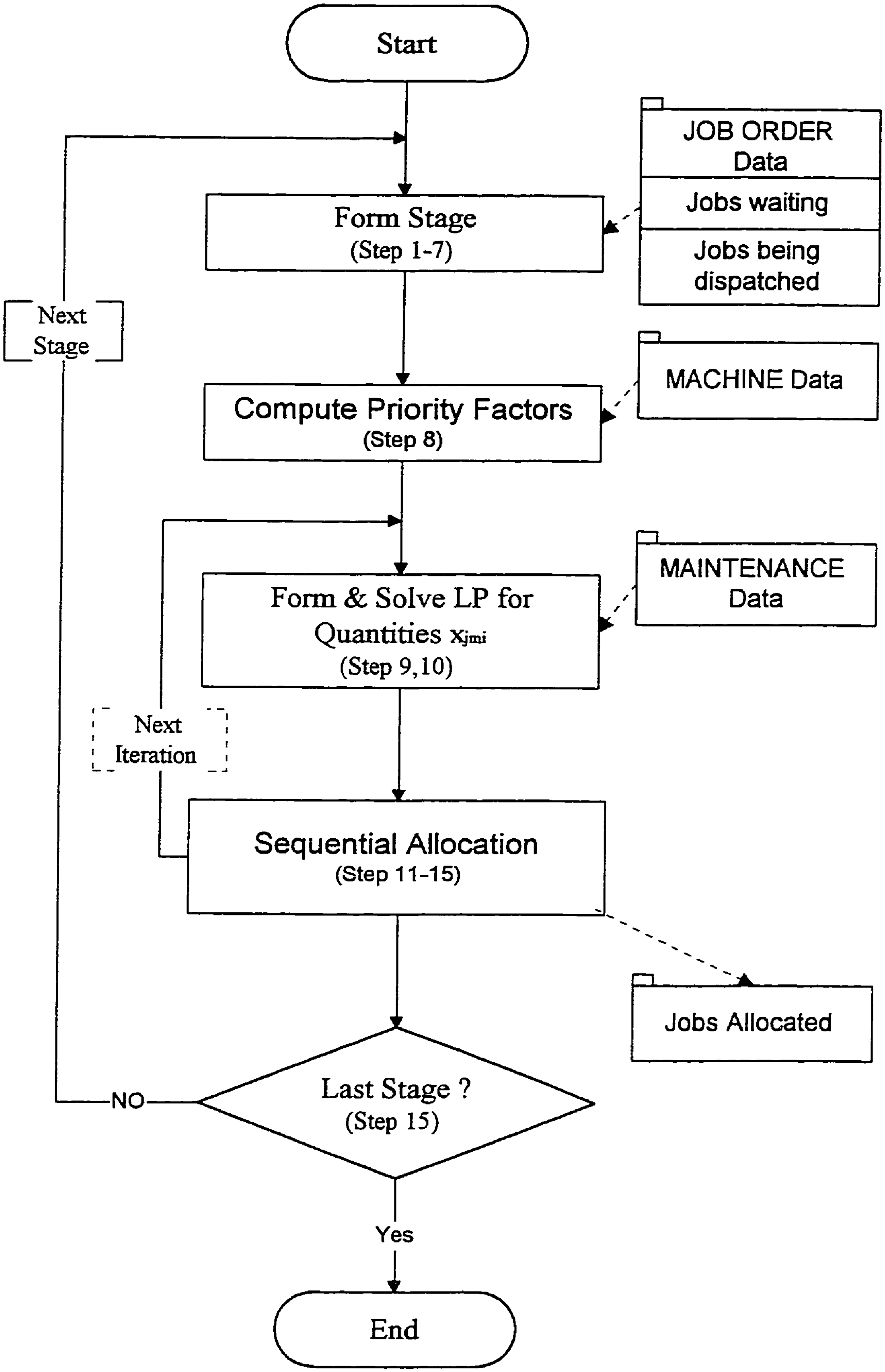

Fig 4.4 Flow chart of the job allocation algorithm

The seven steps essential to form the next stages under consideration are described on Figure 4.5. The dynamic optimization evolves stage by stage. Consequently, the end time of the previous stage becomes the start time of the next. The end time of this stage must abide to the definition of constant job mix stages specified in section 3.2. The end time of the stage will be the earliest time value between: 1)the earliest due time found in the jobs being dispached, or 2) the earliest due time available coming after this stage start time from the list of jobs waiting. Waiting jobs having the same available time as this stage start time will be transferred to the list of jobs being dispatched. The availability of machines varies. Machines undergoing maintenance for the whole stage will not be available. Machines for which maintenance restricts their time available to a portion of the time span covered by the stage will initially be considered available. Other machines will be made fully available. The capacity factor of potential job-machine combinations is enabled at the beginning of a stage. The jobs being dispatched are arranged in increasing order of due dates for computational convenience. This set of operations defines the boundaries and the jobs of a stage.

The computation of priority factors is carried out before proceeding with the allocation of jobs to machines. The priority factors are calculated in accordance with the procedure established in section 3.4.

The secondary loop begins with the LP fomulation. The LP equations created and solved at step 9 conform to the general requirements layed down in section 3.5. Quantities are rounded off to the lower integer at step 10 to ensure that whole quantities will be produced at any given stage independently of the following ones.

Steps 11 to 15 are a detailed breakdown of the sequential allocation process shown on Figure 3.4. Jobs are distributed consecutively, iteration by iteration. The Dispatcher will revise the setups of the job with the earliest due date in the first iteration and save its distribution in the second one. Also, the following job to be allocated will have its setups revised in the second iteration. Jobs have their setups revised and are distributed to the machines by increasing order of due date. The sequential allocation continues until the job with the latest due date has been allocated

The algorithm exits from the last stage when all jobs have been allocated.

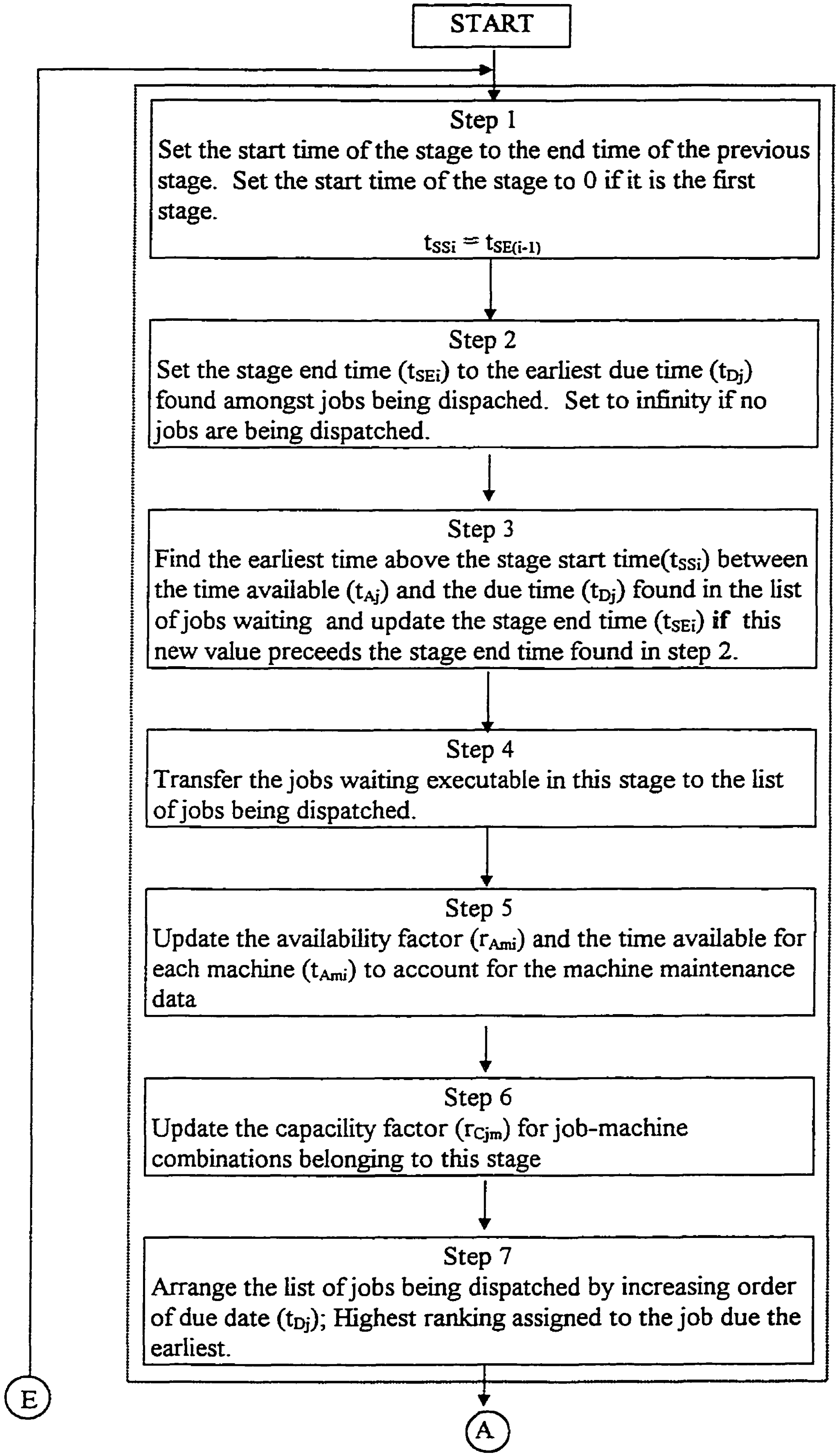

Fig 4.5 Job allocation algorithm

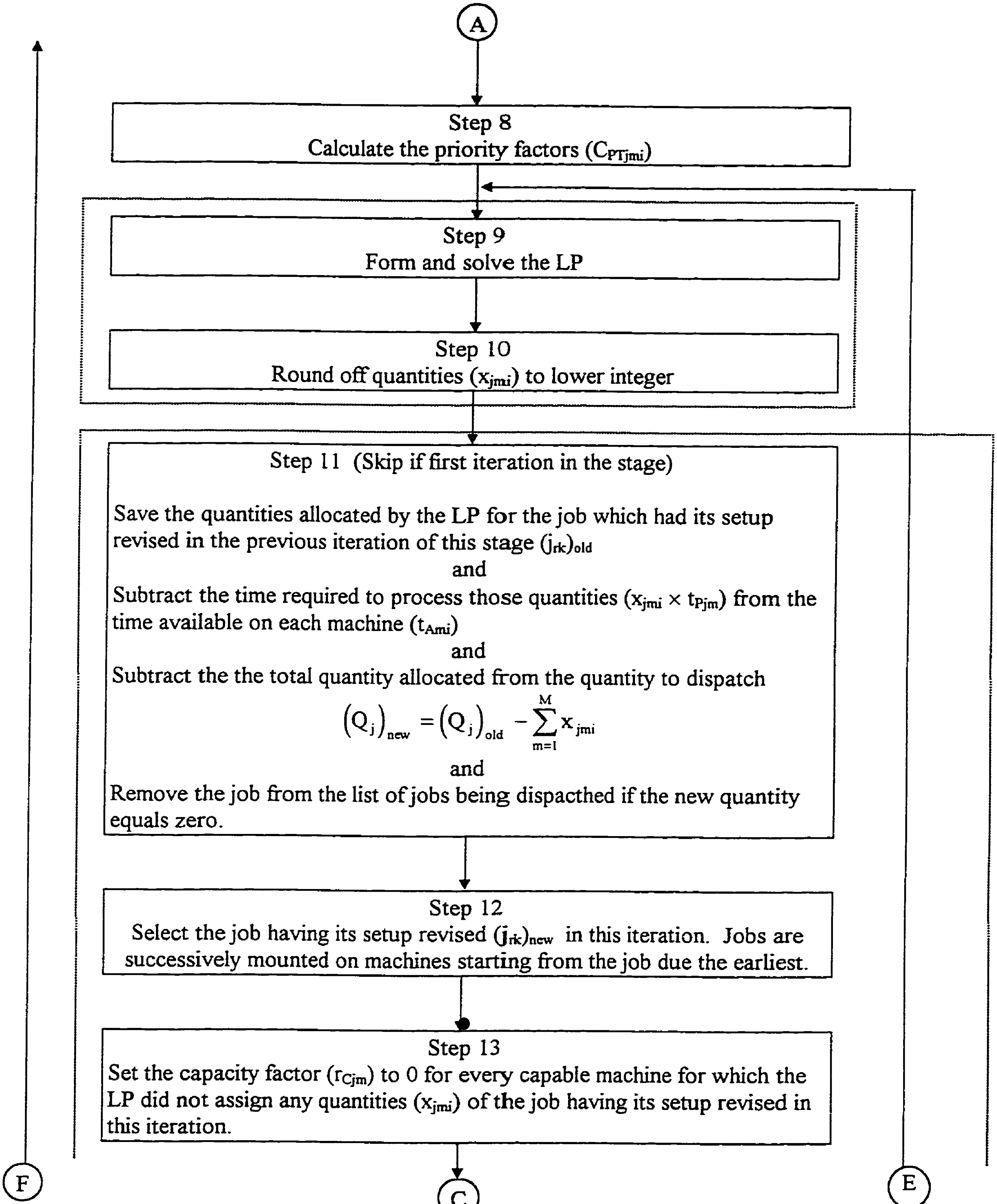

Fig 4.5 Job allocation algorithm (continued)

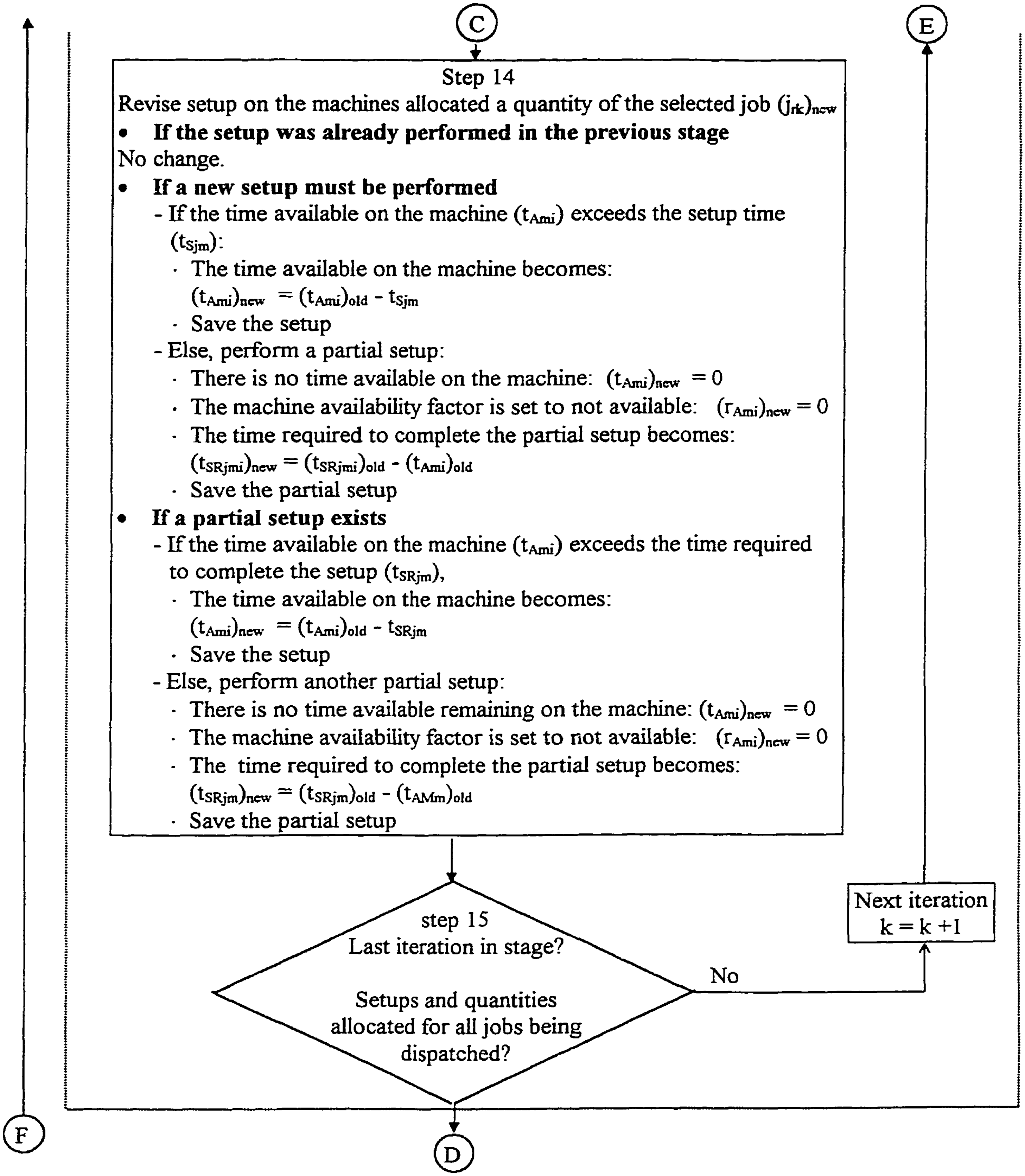

Fig 4.5 Job allocation algorithm (continued)

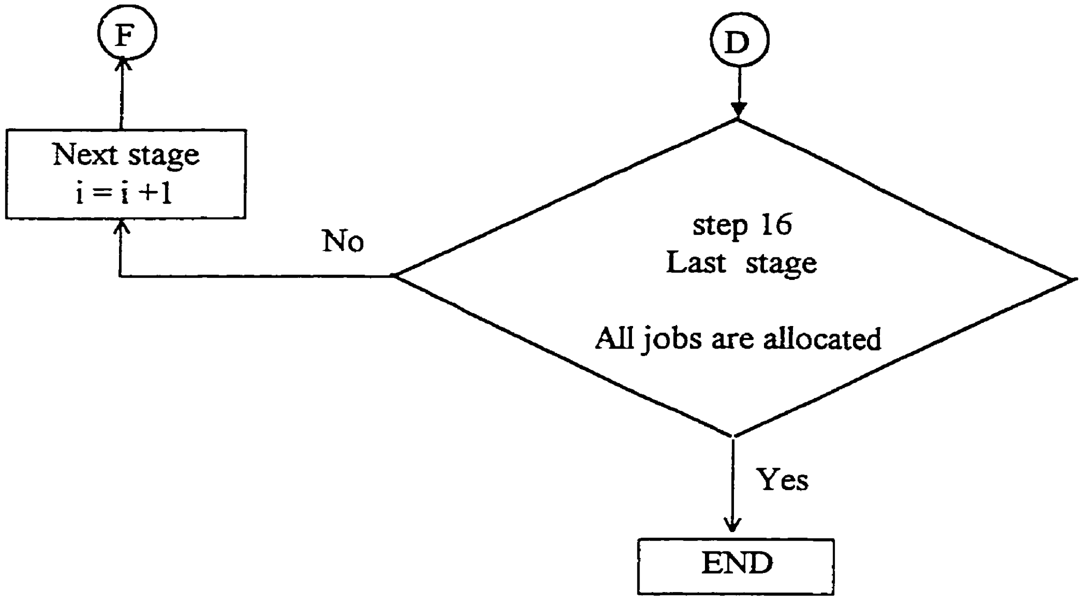

Fig 4.5 Job allocation algorithm (continued)

The allocation algorithm exits from the last stage when all jobs have been allocated.

Chapter 5

Application of the Algorithm

The detailed example presented in this chapter demonstrates the concepts outlined in Chapter 3 and follows the algorithm presented in Chapter 4. The problem is defined and followed by a detailed explanation of the steps involved in allocating jobs for the first stage. The setups and quantities allocated at each stage are summarized in a table.

5.1 Problem Definition

The problem consists in allocating a set of eleven jobs on four machines. Setup times and maintenance periods are also considered in the optimization. A breakdown by constant job mix of the job orders contained in the job orders data of Table 5.1 is presented in Figure 5.1. The dispatcher will not allot any jobs to the machine when they are undergoing maintenance at the time specified in Table 5.2. Table 5.3 and 5.4 respectively contain the setup times and processing times of job orders on the machines.

JOB ORDER Data			
Job	t_{Aj}	t_{Dj}	Q_j
J1	0	960	23
J2	0	960	34
J3	0	960	41
J4	480	960	17
J5	720	1200	13
J6	960	1920	24
J7	960	1920	16
J8	1200	1920	16
J9	1200	1960	44
J10	1200	1960	13
J11	1200	1960	15

Table 5.1 Job order data

MAINTENANCE Data		
M	t_{USm}	t_{UEm}
M3	0	960
M4	960	1510
M1	1610	2160

Table 5.2 Maintenance data

Setup Times (t_{Sjm})				
Job	M1	M2	M3	M4
J1	420	0	396	180
J2	240	472	408	0
J3	0	210	0	186
J4	192	258	174	102
J5	36	42	48	54
J6	216	198	336	192
J7	176	0	144	150
J8	36	0	66	66
J9	0	420	396	0
J10	24	36	24	12
J11	24	48	0	6

Table 5.3 Setup times

Processing Times (t_{Pjm})				
Job	M1	M2	M3	M4
J1	12,00	0	12,80	13,20
J2	20,00	16,00	18,00	0
J3	0	12,00	0	23,50
J4	10,00	6,80	12,00	14,00
J5	10,00	8,90	9,30	8,10
J6	19,00	18,50	12,00	20,00
J7	14,00	0	19,30	19,10
J8	18,00	0	12,00	13,00
J9	0	4,80	13,00	0
J10	21,20	11,75	21,20	14,10
J11	21,40	12,80	0	17,00

Table 5.4 Processing times

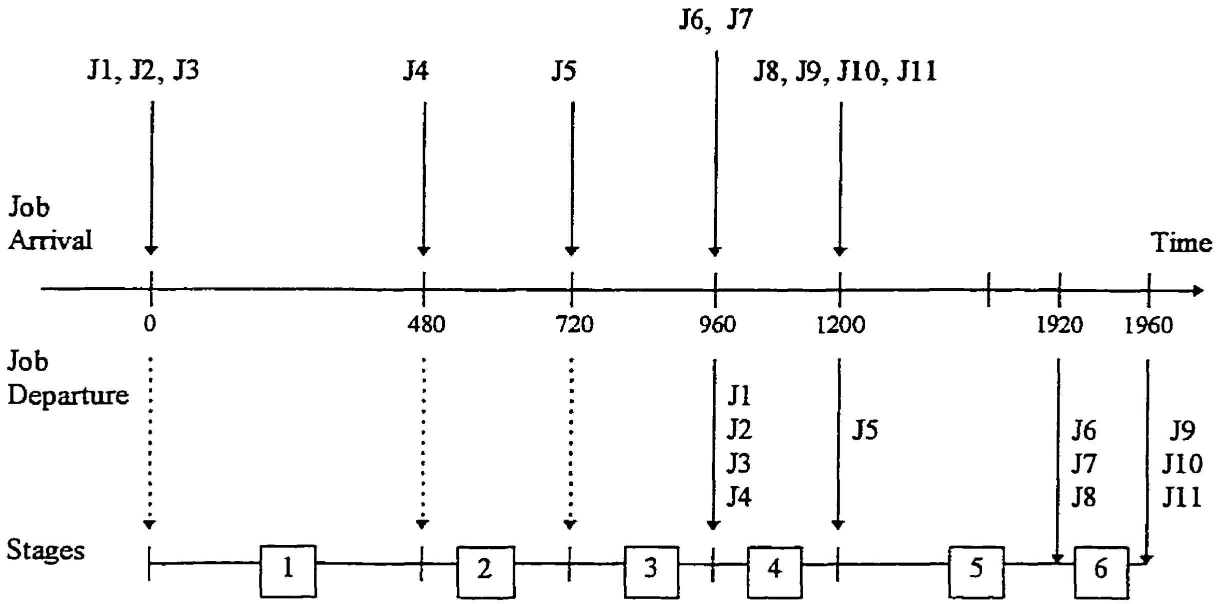

Fig 5.1 Discretization of job orders in constant job mix stages

The Dispatcher allocates the jobs in accordance with the operations mentioned on the flow chart of the job allocation algorithm shown on Figure 4.4. The detailed computations appear on Figure 4.5. Step 1 to 7 govern the formation of the stage based on the job order data. The priority factors will be computed as outlined at step 8. Every iterative loop begins with the quantities to allocate on machines found from step 9 and 10. Then, these quantities are sequentially allocated in steps 11 to 15. The exit from the iterative loop marks the end of the computation for the first stage and the beginning of the next stage.

The following sections reproduce the computations performed in each module of the job allocation algorithm.

5.2 Formation of the 1st Stage

Step 1 to 7 of the job allocation algorithm guide the selection of jobs to be dispatched in the first stage from the eleven jobs waiting. Initially, all jobs are considered as waiting as none have yet been considered by the dispatcher.

- step 1

  Set the start time of the stage to 0 as it is the first stage.

  $t_{SS1} = 0$

- step 2

  The stage end time is set to infinity as there are no jobs being dispatched yet.

  $t_{SE1} = \infty$

- step 3

  The earliest time available later than the stage start time is 480.

  The earliest due time is 960.

  The stage end time found from the jobs being dispatched is ∞

  Hence, set the stage end to 480

  $t_{SE1} = 480$

The stage spans from 0 to 480. These values can represent any units of time as long as they are consistantly used.

- step 4

The executable jobs waiting are transferred to the list of jobs being dispatched.

Jobs being dispatched			
Job	t_{Aj}	t_{Dj}	Q_j
J1	0	960	23
J2	0	960	34
J3	0	960	41

Table 5.5 Jobs being dispatched in the 1st stage

- step 5

The availability factors and time available for each machine are computed from the maintenance data.

m	1	2	3	4
r_{Am1}	1	1	0	1
t_{Am1}	480	480	0	480

Table 5.6 Initial availability factors and time available in 1st stage

- step 6

The capacity factors are updated for all the job/machine combinations.

Capability Factors (r_{Cjm1})				
Job	M1	M2	M3	M4
J1	1	0	1	1
J2	1	1	1	0
J3	0	1	0	1

Table 5.7 Initial capacity factors for stage 1

- step 7

The jobs being dispatched remain in the order shown in Table 5.5.

5.3 Computations of Priority Factors for the 1st Stage

The priority factors are computed in accordance with the equations introduced in section 3.4.

For example, the priority factor for the processing of J1 on machine M1 would be calculated as follows:

$c_{PT\,J1\,M1\,1} = c_{T\,J1\,1} \times c_{P\,J1\,M1\,1}$

where $c_{T\,J1\,1} = (t_{SE1} - t_{SS1}) / (t_{D\,J1} - t_{SS1}) = (480 - 0) / (960 - 0) = 0{,}5$

and $c_{P\,J1\,M1\,1} = (r_{A\,M1\,1})\,(r_{C\,J1\,M1}) / (t_{P\,J1\,M1} + t_{SR\,J1\,M1} / Q_{J1})$

$t_{SR\,J1\,M1} = t_{S\,J1\,M1}$ because the job set on M1 ($j_{S\,M1}$) is not J1

$= (1)\,(1) / (12{,}00 + 420 / 23) = 0{,}03305$

hence

$c_{PT\,J1\,M1\,1} = 0{,}5 \times 0{,}03305 = 0{,}01652$

- step 8

Similar calculations are reproduced for all the job/machine combinations and tabulated.

Priority Factors (C_{PTjmi})				
Job	M1	M2	M3	M4
J1	0,01652	0	0	0,02378
J2	0,01848	0,01673	0	0
J3	0	0,02920	0	0,01783

Table 5.8 Priority factors of the 1st stage

5.4 LP and the Sequential Allocation of the 1st Stage

The iterative loops that performs the sequential allocation begin with the LP formulation. The LP solver yields quantities which are rounded off before setup and quantities are allocated on machines. The loop encompasses step 9 to 15 and continues until all the jobs being dispatched have been considered.

- step 9

The LP equations are formulated and solved.

$$\text{MAX} \quad 0{,}01652\, x_{j1m1} + 0{,}02378\, x_{j1m4} + 0{,}01848\, x_{j2m1} + 0{,}01673\, x_{j2m2} + 0{,}0292\, x_{j3m2} + 0{,}01783\, x_{j3m4}$$

SUBJECT TO

Qj1) $x_{j1m1} + x_{j1m4} \leq 23$

Qj2) $x_{j2m1} + x_{j2m2} \leq 34$

Qj3) $x_{j3m2} + x_{j3m4} \leq 41$

Tm1) $12{,}00\ x_{j1m1} + 20{,}00 x_{j2m1} \leq 480$

Tm2) $16{,}00\ x_{j2m2} + 12{,}00\ x_{j3m2} \leq 480$

Tm4) $13{,}20\ x_{j1m4} + 23{,}50\ x_{j3m4} \leq 480$

$x_{j1m1}, x_{j1m4}, x_{j2m1}, x_{j2m2}, x_{j3m2}, x_{j3m4} \geq 0$

The LP solver proposes quantities to be allocated on machines

		x_{jm1}		
Job	M1	M2	M3	M4
J1	0	0	0	23,00
J2	24,00	4,87	0	0
J3	0	33,49	0	7,50

Table 5.9 Quantities proposed by the LP solver for the first iteration of the 1st stage

- step 10

The quantities are rounded off to the lowest integer

quantities to be allocated on machines

	x_{jm1}			
Job	M1	M2	M3	M4
J1	0	0	0	23
J2	24	4	0	0
J3	0	33	0	7

Table 5.10 Quantities rounded off to the lowest integer

- step 11

Skipped over because it is the first iteration in this stage

- step 12

Job 1 will have its setup revised in this first iteration (k=1) because it has the earliest due date.

$j_{R1} = J1$

- step 13

Setting the capacity factor of machine M1 and M3 to zero because no parts were assigned for the job having its setup revised.

	Capability Factors (r_{Cjm1})			
Job	M1	M2	M3	M4
J1	0	0	0	1
J2	1	1	1	0
J3	0	1	0	1

Table 5.11 Modified capability factors after LP of the 1st iteration

- step 14

The operations for a new setup are performed to process job 1 on machine 4. Setups are revised on M4 only because the LP did not allocate any quantities to the other machines. The setup can be completed and there is some time available remaining to process J1.

$(t_{AM4})_{new} = 480 - 180 = 300$

The time available on machines becomes as shown in the following table. The availability of machines was not modified in the first iteration.

m	M1	M2	M3	M4
r_{Am1}	1	1	0	1
t_{Am1}	480	480	0	300

Table 5.12 Modified availability factors and time available in the 1st stage

- step 15

Proceed to the next iteration as setups and quantities have not been allocated for all the jobs being dispatched.

$k = 1 + 1 = 2$

The second iterative loop begins at step 9

- step 9

The new set of LP equations incorporates the modifications performed in the first iteration.

MAX $0{,}02378\, x_{j1m4} + 0{,}01848\, x_{j2m1} + 0{,}01673\, x_{j2m2} + 0{,}0292\, x_{j3m2} + 0{,}01783\, x_{j3m4}$

SUBJECT TO

Qj1) $x_{j1m4} \leq 23$

Qj2) $x_{j2m1} + x_{j2m2} \leq 34$

Qj3) $x_{j3m2} + x_{j3m4} \leq 41$

Tm1) $20{,}00\, x_{j2m1} \leq 480$

Tm2) $16{,}00\, x_{j2m2} + 12{,}00\, x_{j3m2} \leq 480$

Tm4) $13{,}20\, x_{j1m4} + 23{,}50\, x_{j3m4} \leq 300$

$x_{j1m4}, x_{j2m1}, x_{j2m2}, x_{j3m2}, x_{j3m4} \geq 0$

- step 10

The quantities proposed by the LP solver were rounded off to the lowest integer and are summarized in the following table.

	x_{jml}			
Job	M1	M2	M3	M4
J1	0	0	0	22
J2	24	0	0	0
J3	0	40	0	0

Table 5.13 Quantities of the second iteration rounded off to the lowest integer

- Step 11

The 22 units of job 1 alloted to machine 4 are saved. Job 1 had its setup revised in the previous iteration.

The time required to produce the 22 units is substracted from the time available on machine 4.

$t_{AM41} = 300 - 22 \times 13{,}20 = 9{,}6$

The quantity dispatched is subtracted from the initial quantity.

$(Q_1)_{new} = 23 - 22 = 1$

The jobs remains as a job to be dispatched as there is an extra unit to be produced.

- step 12

Job 2 will have its setup revised in the second iteration

$j_{R2} = J2$

- step 13

The capacity factor of machine 2 and 3 are set to zero because no parts of the job having its setup revised were allocated to them.

	Capability Factors (r_{Cjm1})			
Job	M1	M2	M3	M4
J1	0	0	0	1
J2	1	0	1	0
J3	0	1	0	1

Table 5.14 Modified capability factors after LP of the 2nd iteration

- step 14

Once again, the operations for a new setup are performed. The setup can be completed and there is some time available remaining to process the job having its setup being revised.

$(t_{AM1})_{new} = 480 - 240 = 240$

The time available on machines becomes as shown in the following table. The availability of machines was not modified in the first iteration.

m	M1	M2	M3	M4
r_{Am1}	1	1	0	1
t_{Am1}	480	240	0	9

Table 5.15 Modified availability factors and time available after the 2nd iteration of the 1st stage

- step 15

Proceed to the third iteration as setups and quantities have not been allocated for all the jobs being dispatched.

$k = 2 + 1 = 3$

A third and a fourth iteration will be required to complete the setups and allocate quantities on the machines. The third iteration is required to fix the quantity from job 2 allocated on machine 1 and setup machine 2 for job 3. The quantity processed of job 3 is decided during the fourth iteration.

The following table summarizes the results emerging from the iterations performed in the first stage.

		Stage 1		
Machine	t_{SS1}	t_{SE1}	Job	Setup/Qties
M1	0	240	J2	Setup
	240	480	J2	12 units
M2	0	210	J3	Setup
	210	474	J3	22 units
M3	maintenance			
M4	0	180	J1	Setup
	180	470	J1	22 units

Table 5.16 Job allocation for the 1st stage

- step 16

The algorithm decides whether to move to the next stage or end the job allocation once all the jobs being dispatched have been considered

The algorithm moves to the next stage as some jobs being dispatched and some jobs waiting remain to be allocated.

5.5 Allocations for Each Stage

The Dispatcher performs the same series of computations until all 11 jobs have been allocated to machines. The results of all the six stages of this problem are summarized in Table 5.17. The original results can be found in Appendix A.

	Stage 1	Stage2	Stage 3	Stage 4	Stage 5	Stage 6
Machine	0-480	480-720	720-960	960-1200	1200-1920	1920-2160
M1	0 - 240 setup J2 240-480 12 of J2	480-720 12 of J2	720-920 10 of J2	960-1136 setup J7 1136-1192 4 of J7	1200-1368 12 of J7 1610-1920 maintenance	1920-2160 maintenance
M2	0-210 setup J3 210-474 22 of J3	480-708 19 of J3	720-762 setup J5 762-877 13 of J5	960-1158 setup J6 1158-1195 2 of J6	1200-1620 setup J9 1620-1831 44 of J9 1831-1879 setup J11 1879-1917 3 of J11	1920-1932 1 of J11
M3	maintenance	maintenance	maintenance	960-1200 partial setup J6	1200-1296 setup J6 1296-1560 22 of J6 1560-1626 setup J8 1626-1818 16 of J8	
M4	0-180 setup J1 180-470 22 of J1	480-493 1 of J1 493-595 setup J4 595-707 8 of J4	720-846 9 of J4	maintenance	1200-1510 maintenance 1510-1522 setup J10 1522-1705 13 of J10 1705-1711 setup J11 1711-1898 11 of J11	

Table 5.17 Setup and quantities allocated for the whole problem

Chapter 6

Examples of Parameters Affecting Dispatching

This chapter uses brief examples to demonstrate the influence of priority factors, partial setups and the sequential allocation on dispatching. The value of priority factors is influenced by setup times, quantities to produce and time constraints. Partial setups allow a setup to begin even though the time available in the stage is insufficient to complete the preparation. All the computations follow the algorithm presented in chapter 4.

Job orders data, machine data and maintenance data are input to the job allocation software using text files. The optimization is performed at each stage and the setups and quantity allocated are saved in a new text file. Samples of those files can be found in appendix A. There is no maintenance in any of the examples in order to focus on the precise concept presented in each section.

6.1 Handling of Setup Times

The example shown in this section demonstrates how processing speed based priority factors guide the dispatcher in allocating jobs to the most productive machines. This priority factor consists of the normal processing time plus a fraction of the setup time, corresponding to the initial setup time, divided by the quantity to be allocated. Generally, the smaller the quantity to produce the greater the influence of setup times on the allocation.

The example deals with the case of four jobs that have to be allocated on four machines. They have the same time available and due time, the same quantity to be allocated but different setup and processing times. One stage is sufficient for the allocation. The job orders data and machine data can be found in Appendix A and are summarized in Table 6.1 and 6.2.

The results computed by the dispatcher are presented in Table 6.3. The jobs with the highest processing speed based priority factors are shown in bold in all three tables. The shaded cell indicate the machine which offers the quickest processing time for the job under consideration. If jobs are assigned to the machines shown in the shaded cells, the system would have required about 7.8 % more time (1790 rather than 1660 units of time). Therefore, handling of setup time through priority factors yields a significant reduction in machine time.

Job Order Data			
Job	t_{Aj}	t_{Dj}	Q_j
J1	0	480	20
J2	0	480	20
J3	0	480	20
J4	0	480	20

Table 6.1 Job order data for the handling of setup times

setup times (t_{Sjm})				
	M1	M2	M3	M4
J1	170	120	180	**40**
J2	130	130	**100**	160
J3	**100**	190	140	180
J4	200	**80**	150	60

processing times (t_{Pjm})				
	M1	M2	M3	M4
J1	12,00	13,00	10,00	**15,00**
J2	16,00	16,00	**17,00**	15,00
J3	**15,00**	12,00	14,00	14,00
J4	16,00	**20,00**	18,00	22,00

Table 6.2 Setup and processing times for the handling of setup times

	Stage 1	
Machine	0-480	
M1	0-100	setup **J3**
	100-400	20 of **J3**
M2	0-80	setup **J4**
	80-480	20 of **J4**
M3	0-100	setup **J2**
	100-440	20 of **J2**
M4	0-40	setup **J1**
	40-340	20 of **J1**

Table 6.3 Setup and quantities allocated for the handling of setup times

6.2 Effect of Quantities on Priority Factors

The previous section showed that the algorithm selects the job machine combination with the most productive processing performance based priority factors. Processing performance based priority factors are also influenced by the quantities to be allocated. The example will show that priority factors are influenced not only by the setup times. The quantities to be produced are equally important. The larger the quantities to be allocated, the smaller the influence of setup time on the processing performance based priority factors.

The example demonstrates the allocation of four jobs on four machines. The four jobs share the same time available and due time. Setup and processing time vary. Contrary to the previous example, quantities to be allocated vary between 10 to 30. The job order data and machine data can be found in Appendix A and are summarized in Table 6.4 and 6.5.

Results were compiled by the dispatching software and are summarized in table 6.6. The jobs chosen by the dispatcher are shown in bold in all three tables. The shaded cells indicate the job machine combination that would have been selected if all jobs had 20 units. Allocating jobs to the job-machine combination with the most productive performance based priority factor conveyed a 6,0 % reduction in total time spent on the machines (1510 units of time were necessary rather than 1600). This demonstrates that considering quantities and not only the setup times when selecting the most productive job machine combination improves the machine utilization and shortens the total manufacturing time.

Job Order Data			
Job	t_{Aj}	t_{Dj}	Q_j
J1	0	480	30
J2	0	480	25
J3	0	480	10
J4	0	480	20

Table 6.4 Job order data for the effects of quantities

setup times (t_{Sjm})				
	M1	M2	M3	M4
J1	120	90	270	**120**
J2	**50**	180	180	270
J3	110	**60**	50	20
J4	120	270	**80**	240

processing times (t_{Pjm})				
	M1	M2	M3	M4
J1	9,00	10,00	4,00	**8,00**
J2	**16,00**	12,00	12,00	8,00
J3	12,00	**16,00**	18,00	21,00
J4	14,00	12,00	**20,00**	13,00

Table 6.5 Setup and processing times for the effects of quantities

	Stage 1	
Machine	0-480	
M1	0-50 50-450	setup **J2** 25 of **J2**
M2	0-60 60-220	setup **J3** 10 of **J3**
M3	0-80 80-440	setup **J4** 20 of **J4**
M4	0-120 40-340	setup **J1** 30 of **J1**

Table 6.6 Setup and quantities allocated for the effects of quantities

6.3 Enforcement of Due Dates

The time based priority factors offer a means for handling the extra constraints imposed by the necessity to produce on time. The influence of this priority factor on a given job increases as the stage undergoing dispatching draws near the due time of the job under consideration. Consequently, jobs among those being dispatched with the due time will see the influence of their processing performance base priority factor diminished by the time based priority factor. Jobs nearing the due date will secure more importance.

This example will demonstrate the influence of priority factors by allocating four jobs on four machines. The job order data are shown on Table 6.7. The four jobs have the same time available but different due times. Quantities are fixed at 20 units for all four jobs. Table 6.8 shows that the setup time and the processing time are the same for a particular job on any machine and vary from one job to another. The data input into the allocation system are shown in Appendix A.

The Dispatcher achieved a good compromise, allowing production to be completed on or before time, and confining production to a single stage. The chosen jobs are shown in bold under the respective machine of Table 6.9. The shaded cells represent the best performing job-machine combination. Apart from the first job that had to be allotted to the machine with the fastest processing time, the other jobs were dispatched to the machine that could produce the part the earliest without jeopardizing the due time of others.

Job Order Data			
Job	t_{Aj}	t_{Dj}	Q_j
J1	0	480	20
J2	0	600	20
J3	0	720	20
J4	0	960	20

Table 6.7 Job order data for the enforcement of due dates

setup times (t_{Sjm})				
	M1	M2	M3	M4
J1	**120**	120	120	120
J2	100	**100**	100	100
J3	80	80	**80**	80
J4	60	60	60	**60**

processing times (t_{Pjm})				
	M1	M2	M3	M4
J1	**18,00**	19,00	20,00	21,00
J2	18,00	**19,00**	20,00	21,00
J3	18,00	19,00	**20,00**	21,00
J4	18,00	19,00	20,00	**21,00**

Table 6.8 Setup and processing times for the enforcement of due dates

	Stage 1	
Machine	0-480	
M1	0-120	setup **J1**
	120-480	20 of **J1**
M2	0-100	setup **J2**
	100-480	20 of **J2**
M3	0-80	setup **J3**
	80-480	20 of **J3**
M4	0-60	setup **J4**
	60-480	20 of **J4**

Table 6.9 Setup and Quantities allocated for the enforcement of due dates

6.4 Partial Setups

The example will demonstrate that partial setups augment machine utilization. Partial setups are performed whenever the required setup cannot be completed within the time available or within the remaining time on the machine at a given stage. As the quantities per batches decrease, the number of stages rises and the time span of stages shortens. Shorter stages imply that many setups cannot be started and completed within the same stage. Delaying a setup until a stage has enough time to fully accommodate would unduly lengthens the production. To avoid wasting time, the dispatcher optimally allocates setups to the machines with the highest processing speed and time based priority factor even if the setup cannot be completed in a stage.

The example consists of the five jobs shown in Table 6.10. All jobs are available at the same time and have the same quantity to manufacture. The first job is due within the first stage and has different setup times. The other four jobs must be completed within the second stage. These jobs all have the same setup time. The machine data are shown in Table 6.11. Setups require the same amount of time. All the data generated by the software can be found in Appendix A.

The allocation performed by the Dispatcher is summarized in Table 6.12. The job-machine combinations are emphasized in bold characters. Job 1 was set up and completed in the first stage. Jobs 2 to 5 had a partial setup up allocated to the most performing machine in the first stage. allocated. The partial allocations allowed a 39.4%

saving over a distribution in which setups would have been performed in stages that could fully accommodate them.

Job Order Data			
Job	t_{Aj}	t_{Dj}	Q_j
J1	0	120	20
J2	0	480	20
J3	0	480	20
J4	0	480	20
J5	0	480	20

Table 6.10 Job order data for partial setups

setup times (t_{Sjm})				
	M1	M2	M3	M4
J1	**60**	40	20	10
J2	150	**150**	150	150
J3	150	150	**150**	150
J4	150	150	150	**150**
J5	**150**	150	150	150

processing times (t_{Pjm})				
	M1	M2	M3	M4
J1	**3,00**	6,00	8,00	10,00
J2	18,00	**12,00**	28,00	39,00
J3	39,00	39,00	**13,00**	39,00
J4	45,00	45,00	45,00	**15,00**
J5	**10,00**	20,00	25,00	33,00

Table 6.11 Setup and processing times for partial setups

	Stage 1	Stage 2
Machine	0-120	120-480
M1	0-60 setup **J1** 60-120 20 of **J1**	120-270 setup **J5** 270-470 20 of **J5**
M2	0-120 partial setup **J2**	120-150 setup **J2** 150-390 20 of **J2**
M3	0-120 partial setup **J3**	120-150 setup **J3** 150-410 20 of **J3**
M4	0-120 partial setup **J4**	120-150 setup **J4** 150-450 20 of **J4**

Table 6.12 Setup and quantities allocated for partial setups

6.5 Sequential Allocation

The benefits of a sequential allocation of jobs will be demonstrated through the following example. Setups can be added on machines for which the LP allocated quantities. The time required for the setups was subtracted from the time available on the machine and another LP was solved. Frequently, the jobs for which setup had been performed, had their quantity transferred to another machine by the LP. The setups would be removed and added to the new machine to find that they had returned to the original machine after solving the following LP. All efforts to find a solution were blocked by an infinite loop.

The three job orders for this example are listed in Table 6.13. Table 6.14 contains the machine data. Sequential allocation avoids cycling and leads to the results shown in Table 6.15. Machine swapping did not occur. Job 2 was allocated to M1 and did not revert to M3 even though M3 has the same processing time. Similarly, J3 was allocated to M3 and M4 but did not move back and forth to M1, althought M1 can produce every unit of J3 at the same speed as M3. The input data and the results from the dispatching software can be found in Appendix A.

Sequential allocation of quantities allows partial or complete setups to the most suitable machines and avoids infinite loops. The job machine combinations are selected through priority factors which mirror a commitment to manufacturing deadlines and efficiency.

Job Order Data			
Job	t_{Aj}	t_{Dj}	Q_j
J1	0	120	20
J2	0	480	10
J3	0	480	10

Table 6.13 Job order data for the sequential allocation

setup times (t_{Sjm})				
	M1	M2	M3	M4
J1	40	**20**	40	40
J2	**150**	180	180	150
J3	200	180	**240**	**240**

processing times (t_{Pjm})				
	M1	M2	M3	M4
J1	10,00	**5,00**	10,00	10,00
J2	**10,00**	22,00	10,00	30,00
J3	40,00	60,00	**40,00**	**45,00**

Table 6.14 Setup and processing times for the sequential allocation

	Stage 1	Stage 2
Machine	0-120	120-480
M1	0-120 partial setup **J2**	120-150 setup **J2** 150-250 10 of **J2**
M2	0-20 setup **J1** 20-120 20 of **J1**	
M3	0-120 partial setup **J3**	120-240 setup **J3** 240-480 6 of **J3**
M4	0-120 partial setup **J3**	120-240 setup **J3** 240-420 4 of **J3**

Table 6.15 Setup and quantities allocated for the sequential allocation

Chapter 7

Application of the Job Allocation System

This chapter compares the solution generated from the job allocation system to the results obtained from the allocation by Shortest Operating Time (SOT). The problem definition precedes both allocations and the result analysis.

7.1 Problem Definition

The problem consists in allocating the seven jobs shown in table 7.1 on four machines. Jobs cannot be allotted to the machine 3 that is undergoing maintenance during the time specified in Table 7.2. Table 7.3 and 7.4 respectively contain the setup times and processing times of job orders on the machines.

The job order data and maintenance data for both allocations are found in Appendix A.

Job Order Data			
Job	t_{Aj}	t_{Dj}	Q_j
J1	0	1440	23
J2	0	1440	34
J3	0	1440	41
J4	480	1440	17
J5	480	1920	13
J6	480	1920	24
J7	480	1920	16

Table 7.1 Job order data

Maintenance Data		
M	t_{USm}	t_{UEm}
M3	0	960

Table 7.2 Maintenance data

Setup Times (t_{Sjm})				
Job	M1	M2	M3	M4
J1	130	120	30	0
J2	200	240	0	0
J3	60	160	120	90
J4	30	0	60	80
J5	0	320	200	150
J6	180	100	0	200
J7	0	40	45	150

Table 7.3 Setup times

Processing Times (t_{Pjm})				
Job	M1	M2	M3	M4
J1	26,00	21,00	30,00	0
J2	30,00	25,00	0	0
J3	66,00	51,00	60,00	72,00
J4	8,00	0	10,00	10,00
J5	0	30,00	40,00	50,00
J6	40,00	64,00	0	21,00
J7	0	45,00	42,00	0

Table 7.4 Processing times

7.2 Allocation by the Job Allocation System

The results generated by the job allocation system are summarized in table 7.5 and are reproduced on Figure 7.1. The stages are clearly indicated at the top of the table and the figure.

	Stage 1	Stage 2	Stage 3
Machine	0 - 480	480 - 1440	1440 - 1920
M1	0 - 200 setup of J2 200 - 470 9 of J2	480 - 510 setup of J4 510 - 1438 116 of J4	
M2	0 - 120 setup of J1 120 - 340 10 of J1 330 - 480 partial setup of J2	480 - 570 setup of J2 570 - 1370 28 of J2 1370 - 1410 setup of J7	1440 - 1665 5 of J7
M3	maintenance	480 - 960 maintenance 960 - 1020 setup of J4 1020 - 1100 8 of J4 1100 - 1300 setup of J5 1300 - 1420 3 of J5	
M4	0 - 90 setup of J3 90 - 378 4 of J3	480 - 680 setup of J6 680 - 1436 36 of J6	1440 - 1692 12 of J6

Table 7.5 Setup and quantities allocated by the job allocation system

7.3 Allocation by Shortest Operating Time

Figure 7.1 also shows jobs that were allocated to the machine with the shortest operating time including setup time. The time required to setup and produce each job on each machine with a suitable capability is shown in table 7.6. Jobs appear in bold under the machine on which they were setup. The shaded cells indicate the machine with the fastest processing time when setup times are neglected.

Job	M1	M2	M3	M4
J1	390	330	**310**	0
J2	1430	**1265**	0	0
J3	**324**	364	360	378
J4	**1030**	0	1310	1330
J5	0	410	320	**300**
J6	2100	3172	0	**1208**
J7	0	265	**255**	0

Table 7.6 Total processing times.
Shortest Operating Times including setup time are represented in bold. Fastest processing times are shown in the shaded cell.

7.4 Discussion

This problem demonstrates the effectiveness of the job allocation system over dispatching by the Shortest Operating Time rule.

The SOT methodology, which includes setup time, always allocates jobs to the most productive machine when time constraints do not require multiple machines to process a job. Jobs 1, 3 and 5 would have respectively required an extra 20, 40 and 110 units of manufacturing time if they had been allocated to the machine with the fastest processing time. J6 finishes 58 units of time late when allocated by the SOT technique.

The job allocation system completed all jobs before their due date. J1 is processed on M1 and is completed earlier rather than if it had been allocated to M2 as suggested by the SOT technique. J2 started later due to the production of J1 but finished earlier when produced on M1. J3 was moved to M4 and met its due time. J4 was completed earlier because a few units were processed on M4. J6 was completed much before its due time by allocating J5 to M3 after moving J7 to M2. All jobs were produced before their respective due date and the overall makespan was reduced from the 1988 units of time required by the SOT methodology to 1692 units.

Table 7.7 shows that the job distribution performed by the job allocation system yielded a better machine utilization than the SOT methodology. The enhanced distribution of jobs on machines allowed J6 to be completed before its due date and reduce the makespan by 296 units of time or 14.9%. The additional time spent on setups by the job allocation system was largely compensated by a reduction in idle time and an increase in the average machine utilization from 65.5% to 84.4%. The total processing time was similar for both cases. The gap between certain stages due to rounding off of quantities did not delay production significantly.

All stages	Job Allocation System				SOT with setup included			
	M1	**M2**	**M3**	**M4**	M1	M2	M3	M4
Total Setup Time (units)	230	400	260	290	90	240	75	350
Total Processing Time (units)	1198	1235	200	1296	1264	1025	490	1158
Machine Utilization (%)	84.4	96.6	62.8	93.7	68.1	63.6	55.0	75.9

Table 7.7 Machine utilization

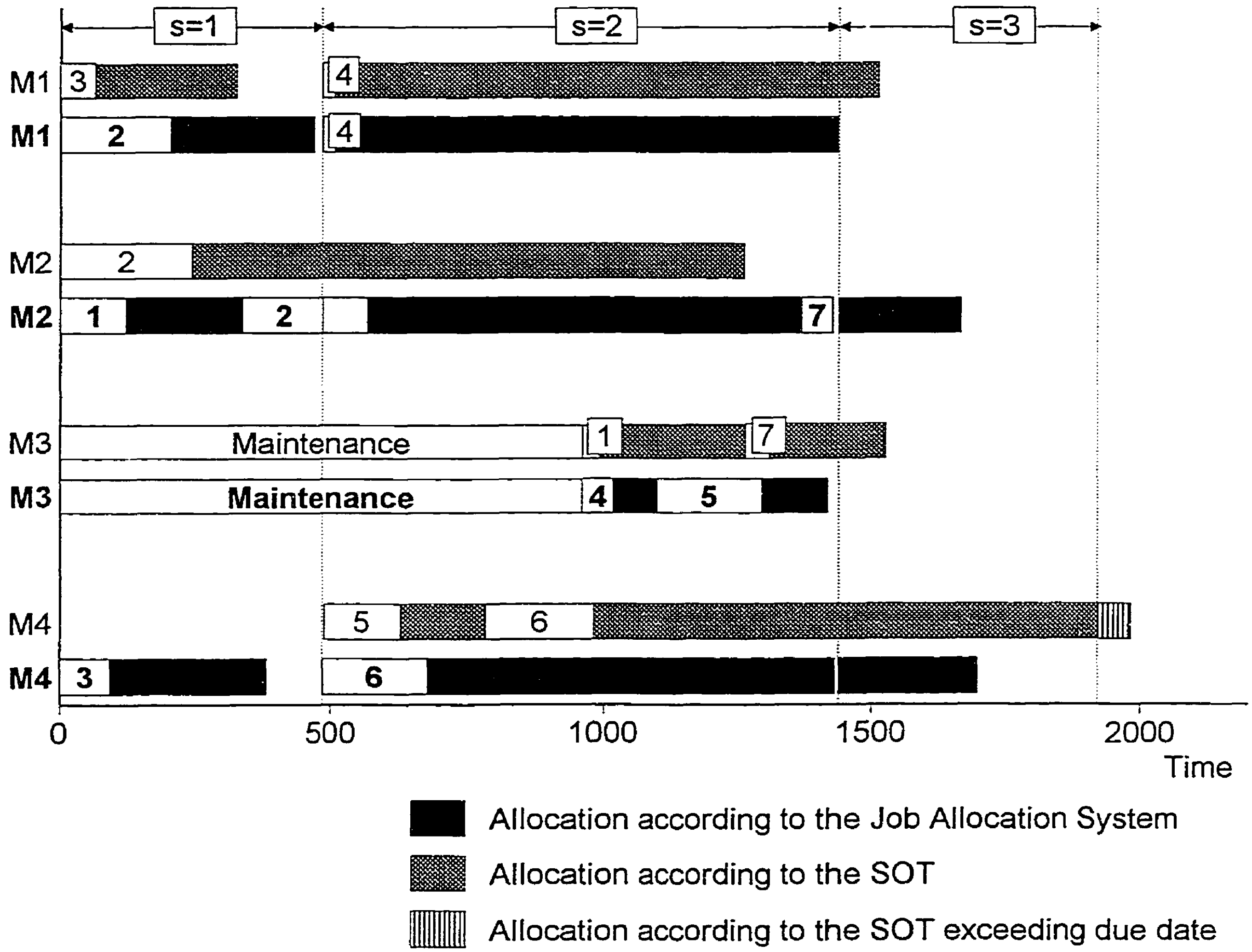

Fig 7.1 Allocation according to the job alloacation system and SOT

Chapter 8

Conclusions and Recommendations for Further Study

8.1 Conclusions

Job allocation on machine tools is one of the most important and difficult task undertaken in the production environment. Research on the subject has produced various ways to handle the task of job allocation. However these proposed solutions have gained only limited acceptance in the manufacturing sphere.

The job allocation system presented in this thesis allocate jobs on machines by performing the computations of a sturdy algorithm that offers numerous benefits:

- The dynamic optimization of job allocation using constant job-mix stages and priority factors yields substantially shorter makespan than the Shortest Operating Time methodology by increasing the machines utilization.

- The discretization of the manufacturing horizon into constant job-mix stages provides unvarying conditions on which dynamic optimization can be applied to maximize the quantity of work performed.

- Priority factors offer an effective mean of guiding the selection of setups by ranking job-machine combinations by processing speed and capacity to meet due times. The necessity for priority factors becomes more important as setup times increase in relation with quantities to be manufactured and the processing times.

- The sequential allocation eliminated infinite loops created by endless setup swapping that occured between machines within a stage.

- The completion of job orders can be speeded up because partial setups allow the preparation of jobs to begin even if the time available in a stage is insufficient to complete set up on the machines.

8.2 Recommendations for Further Study

The benefits of the job allocation scheme presented in this thesis could be enhanced by further pursuing research on the optimization technique used within stages, links between stages and further compression of the schedule by post-processing.

A transition zone could be incorporated at the beginning of a stage to eliminate the delays resulting from the rounding off of quantities. The rounding off of the quantities proposed by the by linear programming at each iteration performed within each stage speeds calculations up and delays the completion of job orders. Rounding off of quantities eliminates the need to use unstable integer programming. On the other hand, the fraction of a piece which cannot be allocated in the stage has to be postponed to the following one. In an extreme case, the job order may not be completed by its due date. Links between stages would be smoother by incorporating a implementing a transition zone at the beginning of a stage. The remaining fraction of an active job order could then be completed earlier without major impacts on the schedule.

Partial setups could also be perfomed on idle machines in the preceding stage of the arrival of a job. This job preemption would also contribute to reduce the makespan.

Post processing optimization techniques such as Tabu Search or the Shifting Bottleneck could be applied to the initial results to further improve machine utilization and shorten the makespan.

Bibliography

[1] Natarajan, S., Dyanamic Optimization of Job Distribution on Machine Tool using Time Decomposition into Constant Job-Mix Stages, M. Eng. Thesis, McGill University, 1992.

[2] Kernighan, Brian A., and Ritchie, Dennis M., The C Programming Language 2nd ed. Prentice Hall, New Jersey, 1988.

[3] Garey, M.R., Johnson, D.S. and Sethi, R., "The complexity of flowshop and jobshop scheduling", Mathematics of Operations Research, 1976, (1), 117-129.

[4] Muth, J.F. and Thompson, G.L., "Industrial Scheduling", Prentice-Hall, New Jersey, 1963.

[5] Markland, Vickery and Davis, "Operations Management", West, 1995.

[6] Powell, N., Production Planning Scheduling and Inventory Control, The Macmillam Company, London, 1970, 108-117.

[7] Nasr, N., and Elsayed, E. A., Job Shop Scheduling with Alternatives Machines, International Journal of Production Research, 1990, 28, 1595-1609.

[8] Kops, L. and Natarajan, S., A Schedule-Based Optimum Allocation of Jobs on Machine Tools, Proceedings CSME Forum, 1992, Vol. 2.

[9] Kops, L. and Natarajan, S., Dynamic Optimization of Job Distribution on Machine Tools Using Time Decomposition into Constant Job-Mix Stages, Annnals of the CIRP, 1992, Vol. 41 (1).

[10] Balazinski, M., and Kops, L., Intelligent Job Allocation on Machine Tools Using Fuzzy Logic, Proceedings of the First World Congress on Intelligent Manufacturing, Puerto Rico, 1995.

[11] Bouchon-Meunier, B., La Logique Floue, Que Sais-Je, Presse Universitaire de France, 1993.

[12] Bugnon, B., Stoffel, K. and Widmer, M., "FUN: A dynamic method for scheduling problems", European Journal of Operational Research, 1995, 83, 271-282.

[13] Reeves, C., "Theory and Methodology, Heuristics for scheduling a single machine subject to unequal job release times", European Journal of Operational Research, 1985, 80, 397-403.

[14] McMahon, G., Florian, M., "On Scheduling with Ready Times and Due Dates to Minimize Maximum Lateness, Operational Research, 1975, Vol. 23, no. 3, May-June, 475-482.

[15] Carlier, J., "The one-machine sequencing problem", European Journal of Operational Research, 1982, 11, 42-47.

[16] Carlier, J., Pinson, E. "An Algorithm for Solving the Job-Shop Problem", Management Science, 1989, February, Vol. 35, No. 2, 164 - 176.

[17] Potts, C.N., "An Adaptive branching rule for the permutation flow-shop problem", European Journal of Operational Research, 1980, 5, 19-25.

[18] Balas, E., "Machine Sequencing via Disjunctive Graphs: an Implicit Enumeration Algorithm, Operational Research, 1968, 940-957.

[19] Adams, J., Balas, E., Zawack, D., "The Shifting Bottleneck Procedure for Job Shop Scheduling", Management Science, 1988, March, Vol. 34, No. 3, 391-401.

[20] Ivens, P. and Lambrecht, M., "Extending the shifting bottleneck procedure to real-life application", European Journal of Operational Research, 1996, 90, 252-268.

[21] Glover, F. "Tabu Search - Part 1", ORSA Journal of Computing, , 1989, Summer, Vol. 1, No. 3, 190-206.

[22] Glover, F. Taillard, E., de Werra, D. "A user`s guide to tabu", Annals of Operations Research, 1993, no. 41, 3-28.

[23] Laguna, M. Velarde, J.L.G. "A search heuristic for just-in-time scheduling in parallel machine", Journal of Intelligent Manufacturing, 1991, no. 2, 253-260.

[24] Dake, S. Batta, R., Lin, L., "Effective Job Shop Scheduling through Active Chain Manipulation", Computers Operationals Research", 1995, Vol. 22, No. 2, 159-172.

[25] Glover, F., Kelly, J.P. and Laguna, M., "Genetic Algorithms and Tabu Search: hybrids for optimization", Computers Operationals Research", 1994, vol. 22, No. 1, 111-134.

[26] Li, S. and Tang, G., "Single Machine Scheduling with Controllable Processing Time: Considering Tardy and Crash Costs", Research Working Paper, McGill University, 1994.

[27] Brandimarte, P., and al., "Continuous Models of Manufacturing Systems, a Review", Annals of the CIRP, 1996, Vol. 45 (1).

[28] Markus, A. and al., "A Maket Approach to Holonic Manufacturing", Annals of the CIRP, 1996, Vol. 45 (1).

[29] Bjorndal, M.H. and al., Some thoughts on combinatorial optimisation. European Journal of Operational Research, 1995, 83, 253-270.

[30] Lourenço, H.R., Job-shop scheduling: Computational study of local search and large-step optimization methods. European Journal of Operational Research, 1995, 83, 347-364.

[31] Blazewicz, J., and al., "Invited Review, The job shop scheduling problem: Conventional and new solution techniques", European Journal of Operational Research, 1996, 93, 1-33.

[32] Reeves, C. R., "Modern Heuristic Techniques for Combinatorial Problems", Blackewell Scientific Publications, Oxford, 1993.

[33] Kops, L. and Natarajan, S., Dynamic Job Allocation on Workstations with CIM-Oriented Schedule-Based Optimization, McGill University, International Industrial Engineering Conference Proceedings, 1992.

[34] Bradley, S.P., Hax, A. C. and Magnanti, T.L., Applied Mathematical Programming, Addison-Wesley Publishing Co. Reading massachusetts, 1977.

[35] Alaee, M., Efficient Loading of Machine Tools in Intermittent Manufacturing M.Eng. Project Report, McGill University, Montreal, Canada, 1978.

[36] Kops, L. and Natarajan, S., Time Partitioning Based on the Job-Flow Schedule - a New Approach to Optimum Allocation of Jobs on Machine Tools, International Journal of Advanced Manufacturing Technology, Springer-Verlag, 1994, pp. 204-210.

[37] Foulds, L.R. , Combinatorial Optimization for Undergraduates, Undergraduate Texts in Mathematics, Springer-Verlag, New York, 1984.

[38] Hiller, F. S., and Lieberman, G.J.,1990, Introduction to Operations Research, 5th Edition, McGraw-Hill Publishing Company, New York.

[39] Philips, D.T., and Ravindran, A. Solberg, J.J., Operations Research and Practice. John Wiley & Sons Inc., New York, 1976.

Appendix A

Job order data and machine data for the example of Chapter 4

Setup File Data
File Name: sb27i.suf
Date: 96/11/27

Total Machine Number = 4
Machine List:
A: TL
B: EL
C: GR
D: MM

Total Job Number = 11

Job1:	0	960	23	0
Job2:	0	960	34	0
Job3:	0	960	41	0
Job4:	480	960	17	0
Job5:	720	1200	13	0
Job6:	960	1920	24	0
Job7:	960	1920	16	0
Job8:	1200	1920	16	0
Job9:	1200	2160	44	0
Job10:	1200	2160	13	0
Job11:	1200	2160	15	0

Setup Time Table:

420	0	396	180
240	472	408	0
0	210	0	186
192	258	174	102
36	42	48	54
216	198	336	192
176	0	144	150
36	0	66	66
0	420	396	0
24	36	24	12
24	48	0	6

Processing Time Table:

12.00	0	12.80	13.20
20.00	16.00	18.00	0
0	12.00	0	23.50
10.00	6.80	12.00	14.00
10.00	8.90	9.30	8.10
19.00	18.50	12.00	20.00
14.00	0	19.30	19.10
18.00	0	12.00	13.00
0	4.80	13.00	0
21.20	11.75	21.20	14.10
21.40	12.80	0	17.00

Maintenance data for the example of Chapter 4

Machine Unavailability File
File Name: cb27a.ctg
Date: 96/11/27

Machine Unavailability List:

3:	0	960	0
4:	960	1510	0
1:	1610	2160	0
0:	0	0	0

Results for the example of Chapter 4

Job Allocation
Date: 1996/11/27

Setup File: sb27j.suf
Machine Contingency File: cb27a.ctg
Job Allocation File: j303all.jaf

Machine Job Allocation:
(M st dt J qty)
1 0 240 s2 0
1 240 480 2 12
2 0 210 s3 0
2 210 474 3 22
4 0 180 s1 0
4 180 470 1 22
1 480 720 2 12
2 480 708 3 19
4 480 493 1 1
4 493 595 s4 0
4 595 707 4 8
1 720 920 2 10
2 720 762 s5 0
2 762 877 5 13
4 720 846 4 9
1 960 1136 s7 0
1 1136 1192 7 4
2 960 1158 s6 0
2 1158 1195 6 2
3 960 1200 ss6 0
1 1200 1368 7 12
2 1200 1620 s9 0
2 1620 1831 9 44
2 1831 1879 s11 0
2 1879 1917 11 3
3 1200 1296 s6 0
3 1296 1560 6 22
3 1560 1626 s8 0
3 1626 1818 8 16
4 1510 1522 s10 0
4 1522 1705 10 13
4 1705 1711 s11 0
4 1711 1898 11 11
2 1920 1932 11 1

Sample data for handling of setup times

Setup File Data
File Name: s970308a.suf
Date: 97/03/08

Total Machine Number = 4
Machine List:
A: TL
B: EL
C: GR
D: MM

Total Job Number = 4
Job1: 0 480 20 0
Job2: 0 480 20 0
Job3: 0 480 20 0
Job4: 0 480 20 0

Setup Time Table:

170	120	180	40
130	130	100	160
100	190	140	180
200	80	150	60

Processing Time Table:

12.00	13.00	10.00	15.00
16.00	16.00	17.00	15.00
15.00	12.00	14.00	14.00
16.00	20.00	18.00	22.00

Job Allocation
Date: 1997/3/8

Setup File: s970308a.suf
Machine Contingency File: null.ctg
Job Allocation File: j970308a.jaf

Machine Job Allocation:
(M st dt J qty)
1 0 100 s3 0
1 100 400 3 20
2 0 80 s4 0
2 80 480 4 20
3 0 100 s2 0
3 100 440 2 20
4 0 40 s1 0
4 40 340 1 20

Sample data for the effects of Quantities on Priority Factors

Setup File Data
File Name: s970308b.suf
Date: 97/03/08

Total Machine Number = 4
Machine List:
A: TL
B: EL
C: GR
D: MM

Total Job Number = 4

Job1:	0	480	30	0
Job2:	0	480	25	0
Job3:	0	480	10	0
Job4:	0	480	20	0

Setup Time Table:

120	90	270	120
50	180	180	270
110	60	50	20
220	270	80	240

Processing Time Table:

9.00	10.00	4.00	8.00
16.00	12.00	12.00	8.00
12.00	16.00	18.00	21.00
14.00	12.00	20.00	13.00

Job Allocation
Date: 1997/3/8

Setup File: s970308b.suf
Machine Contingency File: null.ctg
Job Allocation File: j970308b.jaf

Machine Job Allocation:
(M st dt J qty)
1 0 50 s2 0
1 50 450 2 25
2 0 60 s3 0
2 60 220 3 10
3 0 80 s4 0
3 80 480 4 20
4 0 120 s1 0
4 120 360 1 30

Sample data for the enforcement of the due dates

Setup File Data
File Name: s970308c.suf
Date: 97/03/08

Total Machine Number = 4
Machine List:
A: TL
B: EL
C: GR
D: MM

Total Job Number = 4

Job1:	0	480	30	0
Job2:	0	480	25	0
Job3:	0	480	10	0
Job4:	0	480	20	0

Setup Time Table:

120	90	270	120
100	180	180	270
110	60	50	20
140	200	80	180

Processing Time Table:

9.00	10.00	4.00	8.00
16.00	12.00	12.00	8.00
12.00	16.00	18.00	21.00
14.00	12.00	20.00	13.00

Job Allocation
Date: 1997/3/8

Setup File: s970308c.suf
Machine Contingency File: null.ctg
Job Allocation File: j970308c.jaf

Machine Job Allocation:
(M st dt J qty)
1 0 140 s4 0
1 140 420 4 20
2 0 180 s2 0
2 180 480 2 25
3 0 50 s3 0
3 50 230 3 10
4 0 120 s1 0
4 120 360 1 30
4 360 480 ss2 0

Sample data for partial setups

Setup File Data
File Name: s970308d.suf
Date: 97/03/08

Total Machine Number = 4
Machine List:
A: TL
B: EL
C: GR
D: MM

Total Job Number = 4

Job1:	0	480	30	0
Job2:	0	480	25	0
Job3:	0	480	10	0
Job4:	0	480	20	0

Setup Time Table:

120	90	270	120
100	220	220	270
110	60	50	20
140	200	80	180

Processing Time Table:

9.00	10.00	4.00	8.00
16.00	12.00	12.00	8.00
12.00	16.00	18.00	21.00
14.00	12.00	20.00	13.00

Job Allocation
Date: 1997/3/8

Setup File: s970308c.suf
Machine Contingency File: null.ctg
Job Allocation File: j970308c.jaf

Machine Job Allocation:
(M st dt J qty)
1 0 140 s4 0
1 140 420 4 20
2 0 180 s2 0
2 180 480 2 25
3 0 50 s3 0
3 50 230 3 10
4 0 120 s1 0
4 120 360 1 30
4 360 480 ss2 0

Sample data for the benefits of a sequential allocation

Setup File Data
File Name: s970309e.suf
Date: 97/03/09

Total Machine Number = 4
Machine List:
A: TL
B: EL
C: GR
D: MM

Total Job Number = 3

Job1:	0	120	20	0
Job2:	0	480	10	0
Job3:	0	480	10	0

Setup Time Table:

40	20	40	40
150	180	180	150
200	180	240	240

Processing Time Table:

10.00	5.00	10.00	10.00
10.00	22.00	10.00	30.00
40.00	60.00	40.00	45.00

Job Allocation
Date: 1997/3/8

Setup File: s970308e.suf
Machine Contingency File: null.ctg
Job Allocation File: j970308e.jaf

Machine Job Allocation:
(M st dt J qty)
1 0 50 s2 0
1 50 450 2 25
1 450 480 ss4 0
2 0 60 s3 0
2 60 220 3 10
2 220 420 s4 0
2 420 480 4 5
4 0 120 s1 0
4 120 360 1 30
4 360 480 ss2 0

Job order data and machine data for the example of Chapter 7

Setup File Data
File Name: sch7.suf
Date: 97/05/05

Total Machine Number = 4
Machine List:
A: TL
B: EL
C: GR
D: MM

Total Job Number = 7

Job1:	0	1440	10	0
Job2:	0	1440	41	0
Job3:	0	1440	4	0
Job4:	480	1440	125	0
Job5:	480	1920	3	0
Job6:	480	1920	48	0
Job7:	480	1920	5	0

Setup Time Table:

130	120	0	0
20	240	0	0
60	160	120	90
30	0	60	80
0	320	200	150
180	100	0	200
0	40	45	0

Processing Time Table:

2600	2100	30.00	0
30.00	25.00	0	0
66.00	51.00	60.00	72.00
8.00	0	10.00	10.00
0	30.00	40.00	50.00
40.00	64.00	0	21.00
0	45.00	42.00	0

Maintenance data for the example of Chapter 7

Machine Unavailability File
File Name: cch7.ctg
Date: 97/04/17

Machine Unavailability List:

3:	0	960	0
0:	0	0	0

Results for the example of Chapter 7

Job Allocation
Date: 1997/5/5

Setup File: sch7.suf
Machine Contingency File: cch7.ctg
Job Allocation File: jch7.jaf

Machine Job Allocation:
(M st dt J qty)

1	0	200	S2	0
1	200	470	2	9
2	0	120	s1	0
2	120	330	1	10
2	330	480	ss2	0
4	0	90	s3	0
4	90	378	3	4
1	480	510	s4	0
1	510	1438	4	116
2	480	570	s2	0
2	570	1370	2	32
2	1370	1410	s7	0
3	960	1020	s4	0
3	1020	1100	4	8
3	1100	1300	s5	0
3	1300	1420	5	3
4	480	680	s6	0
4	680	1436	6	36
2	1440	1665	7	5
4	1440	1692	6	12

IMAGE EVALUATION TEST TARGET (QA-3)

150mm

6″

APPLIED IMAGE . Inc
1653 East Main Street
Rochester, NY 14609 USA
Phone: 716/482-0300
Fax: 716/288-5989