CRANFIELD INSTITUTE OF TECHNOLOGY

SCHOOL OF INDUSTRIAL AND MANUFACTURING SCIENCE
Ph.D. THESIS

Academic Year: 1989-91

DAGOBERTO ALVES DE ALMEIDA

CRITICAL PATH ANALYSIS TYPE SCHEDULING IN A FINITE CAPACITY ENVIRONMENT

Supervisors: P.E. Galgut
John M. Kay

September 1992

This thesis is submitted in partial fulfilment for the degree of Doctor of Philosophy

ACKNOWLEDGEMENTS

I wish to express my sincere gratitude to Mr Paul E. Galgut, a fine gentleman, for the invaluable advice, patience and continuous encouragement throughout the course of this research programme.

My very special thanks to Dr John Kay, who with his generosity, good humour and indispensable guidance helped me along these research years.

My gratitude to the staff of building 30. Without their help everything would be much more difficulty.

My thanks are due to *Coordenação de Aperfeiçoamento de Pessoal de Nível Superior* - **CAPES - Brazil** and to *Escola Federal de Engenharia de Itajubá* - **EFEI**, for providing me with the opportunity to further my studies at Cranfield Institute of Technology.

Thanks Zeyad, your friendship gave me the additional push I needed.
To all my friends, so many to name, for their friendship and support.

Finally, my very special thanks are due to my wife whose love, care and unwavering support have made possible the fruition of this work and to my children, Daniel and Camila, for the joys of life.

ABSTRACT

In order to cope with more realistic production scenarios, scheduling theory has been increasingly considering assembly job shops. Such an effort has raised synchronization of operations and components as a major scheduling issue. Most effective priority rules designed for assembly shops have incorporated measures to improve coordination when scheduling assembly structures. However, by assuming a forward loading, the priority rules designed by these studies schedule all operations as soon as possible, which often leads to an increase of the work-in-progress level.

This study is based on the assumption that synchronization may be improved by sequencing rules that incorporate measures to cope with the complexity of product structures. Moreover, this study favours the idea that, in order to improve synchronization and, consequently, reduce waiting time, backward loading should be considered as well. By recognizing that assembly shop structures are intrinsically networks, this study investigates the feasibility of adopting the Critical Path Method as a sequencing rule for assembly shop. Furthermore, since a Critical Path type scheduling requires a precise determination of production capacity, this study also includes Finite Capacity as a requisite for developing feasible schedules.

In order to test the above assumptions, a proven and effective sequencing rule is selected to act as a benchmark and a simulation model is developed. The simulation results from several experiments showed significant reduction on the waiting time performance measure due to the adoption of the proposed critical path type priority rule.

Finally, a heuristic procedure is proposed as a guideline for designing scheduling systems which incorporate Critical Path based rules and Finite Capacity approach.

CONTENTS Page

CONTENTS Page

CONTENTS Page

APPENDICES Page

Tables page

Tables page

CHAPTER 1

Introduction

This chapter discusses the potential of adopting network theory into production scheduling. Particularly, the viability of embracing the Critical Path Analysis Method for sequencing operations of assembly jobs. Some conceptual considerations adopted by this study are featured, as well as the objectives of the thesis. Finally, the basic steps of the methodology employed are given.

1.1. BACKGROUND

In spite of the emphasis given by the production scheduling research to string type jobs where no assembly operation takes place, assembly shop scheduling has been increasingly investigated over the years. The fact that it does represent a larger multitude of jobs than the single component job shop is one of the factors justifying the growing interest from the academic circle **(Philipoom et al., 1991)**. Moreover, assembly shop scheduling provides valuable insights in terms of considering the production process as a whole. The emphasis given to string type jobs is justifiable by the following reasons: (i) the breaking of the overall job into its components has the advantage of facilitating the handling of the scheduling problem. Doubtless, such an approach reduces the inherent complexity provoked by the relationship among components; (ii) scheduling isolated components usually means a smaller production period than the one that would be required if the whole product structure were considered. Such an aspect helps to maintain the scheduling integrity for a relatively longer time. However, such an approach also leads to a lack of visibility of the scheduling process as a whole. The reason lies in the fact that multiple jobs are created from the original order and then treated up to their assemblies as individual entities. In this fashion, aspects such as synchronization of components have no need to arise, which may, however, result in increasing the number of intermediate stocks, since the production of parts is generally anticipated. In recent years, especially after the advent of the production system named OPT (Optimized Production Technology), keywords such as synchronization, have become the order of the day. In terms of assembly shop scheduling, synchronization is not only a question of approach, but a necessity due to the need of coordinating operations of parallel components towards common assemblies.

As will be considered in the next chapter, much research has been undertaken recently in order to study the more realistic environment of the assembly job shop. The most effective rules presented in the pertinent literature include measures to provide coordinating features when scheduling parallel operations. As for any priority rules, their sequencing capability attempts to select an adequate operation to be scheduled in a certain machine. Such a procedure is expected to produce a feasible schedule and good results in terms of a pre-defined set of performance measures.

A schedule is said to be unfeasible if any of its routing orderings, within and across jobs, violates the given technological sequence constraints of operations. Accordingly, a feasible schedule is the one which has operations relationships consistent with the given technological constraints.

Despite efforts on the coordination process of components in assembly shops undertaken recently, this study considers that improvements on synchronization may still be accomplished. As it will be presented in the next chapter, the heuristic priority rules designed for assembly shops assume a forward loading approach for all operations of all components of the assembly structure under consideration. As operations are scheduled as soon as possible, such an approach often leads to an increase of the work-in-progress level since production is anticipated. One of the reasons for the success of the *Just In Time* (JIT) system is due to the negation of such a practice. This study assumes that improvements on synchronization of operations and components reduces waiting time and consequently work-in-progress in multi-product, multi-level job shops. The basic assumption, in which this study is based, is that the adoption of *Critical Path*

Analysis method may improve synchronization among operations and components of assembly shops.

The applicability of network theory to scheduling problems has been illustrated through many studies (**Davis, 1973; Hu, 1961; Trilling, 1966; White and Rogers, 1990**). On the other hand, since its origin, network techniques such as Critical Path Method (CPM) and Project Evaluation and Review Technique (PERT) have been seen as having limited use in manufacturing activities. In opposition, **Miller (1962)** states " PERT can be, and has been, used very effectively through the preliminary manufacturing phases of production prototype or pilot model construction and in the assembly and test of final production equipment which are still 'high on the learning curve'." However, as a scheduling procedure for a job shop, Critical Path Analysis (CPA) has had rather limited application. **Heuser and Wynne (1963)** describe a CPA usage in a medium-sized tool shop. Even though backward and forward passes are an integral element of the CPA theory, its applicability to the sequencing problem has not been considered mainly due to the string type job structures, where backward and forward passes are excluding options for the scheduling process. That is, either scheduling is carried out in backward manner from the planned due-date or forward from a given launch date. In the latter case, the planned due date is checked against the calculated final completion time. On the other hand, if project scheduling differs from assembly, from a managerial point of view (**Davis, 1973**), assembly shop scheduling may be formulated in such a way as to emphasise their network similarities. Such a similarity would suggest the use of backward and forward passes in assembly shop problems based upon the product's critical path. As in single project scheduling, the critical path could be scheduled in the forward manner and the non-critical paths would thus be scheduled in the backward manner.

The forward-backward approach through the critical path of a single project provides the ultimate rule in terms of synchronization, i.e., an operation is scheduled to start at a date, which allows its completion just when required. Nevertheless, with many single projects competing for scarce resources, such an advantage may be partially reduced or even totally lost. If instead of a project structure, the problem was related to an assembly structure, the effect would be similar.

Another inherent difficulty of scheduling in a backward manner, refers to the risk of scheduling a certain operation to start before the launch date. This problem increases in an assembly shop, with many components sharing different resources in different routings.

1.2. CONCEPTUAL CONSIDERATIONS

This section is related to terms, classifications and concepts which will be used throughout this study.

1.2.1. TERMS

(i) String type jobs in which the job consists of just one component manufactured through serial operations and no assembly operation takes place, are related to a *job shop*.
(ii) Multi-level assembly jobs, or simply assembly jobs, in which components include both serial and parallel operations and assemblies take place in different structure levels are related to an *assembly shop*.
(iii) Parallel operations are those performed on components which belong to the same assembly.
(iv) Staging time is the delay encountered by components coming to an assembly point when they have to wait for one another before their assembly operation can start **(Adam et al., 1987)**. Staging time is a performance measure specially designed for assembly shops.
(v) The word *Synchronization* became quite fashionable with the advent of the production system approach called *Optimized Production Technology* **(OPT)** in the early 1980s. For OPT, operations performed on non-bottleneck resources must be synchronized with operations carried out on the bottleneck resource,

which then sets the pace for all the related activities[1]. The assembly shop scheduling theory sees synchronization in a rather more specific manner. Synchronization, or *pacing* as it is called by **Siegel (1971)**, is associated with the coordination of assembly components. Synchronization is obtained by assigning a priority value to a component in relation to the progress of other immediately related components. Therefore, synchronization is seen as an integral element of an assembly sequencing rule. This study regards synchronization as in the latter way.

1.2.2. PRODUCT STRUCTURE

In this study, product structures are represented by three different types of Bill of Materials (BOM), namely *Flat, Tall* and *Mixed.* Mostly research on assembly shop sequencing rules have adopted a similar classification, e.g. **Fry et al., (1989)**. In figure 1.1, BOMs 1 and 2 represent Flat structures, BOMs 3 and 4 represent Tall structures and BOMs 5 and 6 represent Mixed structure which have characteristics of both the Flat and Tall types. The squares represent a component.

As an attempt at formalizing the classification of BOMs it could be said that (i) a Flat structure is characterized by having two level and at least two components per level, (ii) a Tall structure has more than two levels and two components per level, and finally (iii) a Mixed structure is the one which has more than two level and more than two components in at least one level. Such a classification agrees with most research. Accordingly, a string type job is classified as having one level and one component only.

1.2.3. CAPACITY

Project scheduling under resource constraints basically regards capacity in terms

---

[1]: Additional details are given in chapter 2, Literature Review.

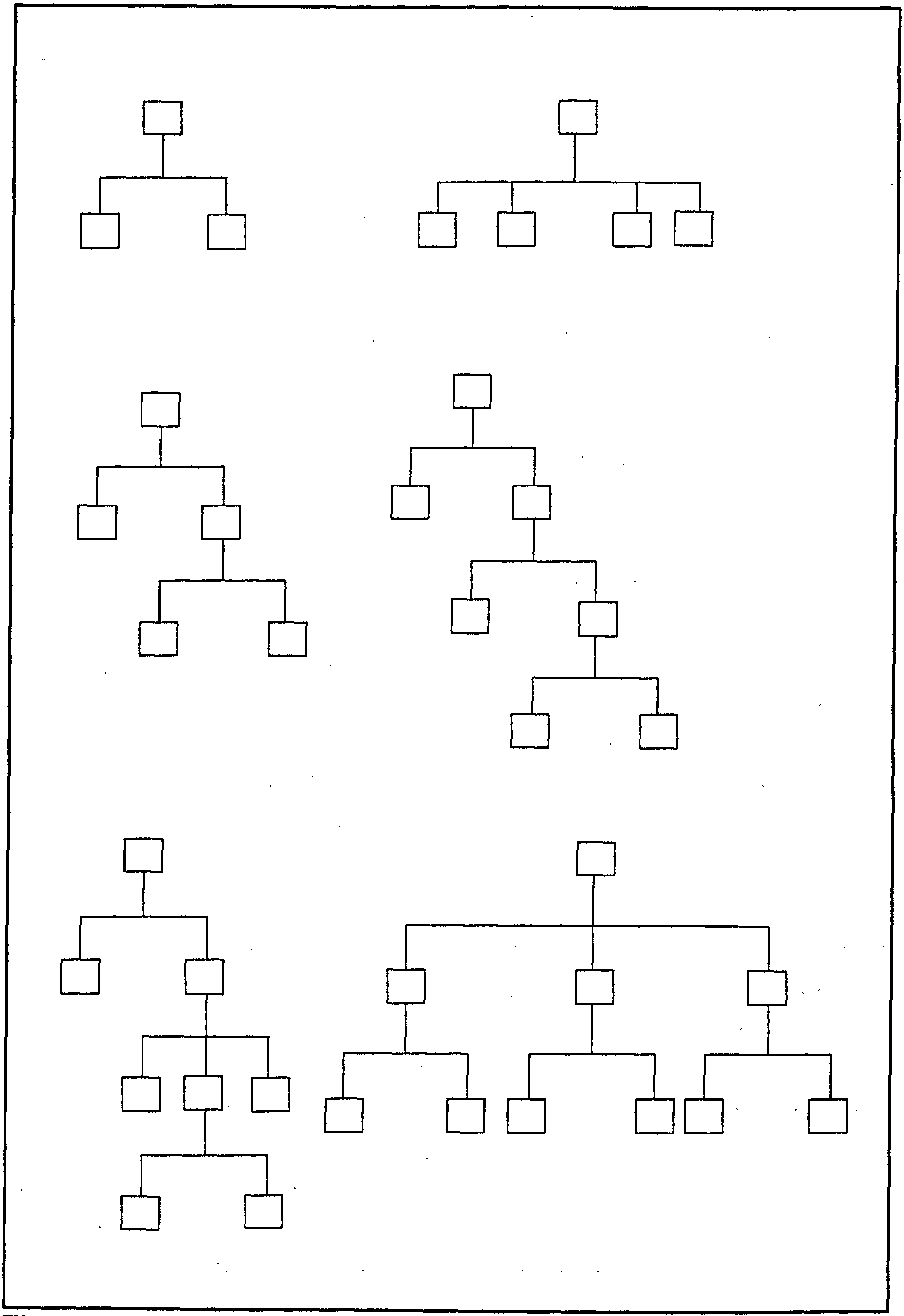


Figure 1.1: Typical BOMs

of the ratio between load and availability level within a given time constraint. The schedule is accomplished by smoothing the resource profile through the float available to the activity in analysis. If necessary the time constraint is extended ahead in time (**Battersby, 1970**). Many heuristic methods have been suggested for smoothing resources, thus providing reasonably good, but not necessarily optimal solutions. Therefore, capacity smoothing and load levelling are similar concepts.

The accomplishment of the research objective requires a new approach to the way available capacity is usually seen in the pertinent literature. After all, by scheduling in the forward manner it is assumed that there always exist available capacity ahead in time. Backward scheduling provides an additional difficulty to the scheduling process once there is an risk of scheduling an operation before the order launch date, or even to a date already past. Whatever the adopted scheduling approach, forward or backward, the knowledge on available capacity has to be accurate in size and timing. Available capacity, as opposed to load, is related to a time interval at a specific resource, no matter which kind of resource, e.g. labour, tool, machine or work centre. The availability is considered as a window with starting and final limits. Such an assumption is relevant to the scheduling proposal, since the backward pass generates intervals (slots, windows) of capacity availabilities. Figure 1.2 describes the relation between load and capacity, which is represented as an availability slot.

Such an approach is pertinent on its own, once it dictates a view on capacity in terms of its constraints, i. e., timing and size. Besides, it is also consistent with the increasing interest on finite capacity scheduling that, thanks to advances in computer technology, have been increasingly adopted. The inclusion of Finite Scheduling is considered to be advantageous to the objectives of this study since it adopts a more realistic view on scheduling by accepting the inherent capacity limitations of a shop floor.

Scheduling is usually seen as a loading task, in the sense that operations are piled up one over another. However, quite often, slots of availabilities are generated

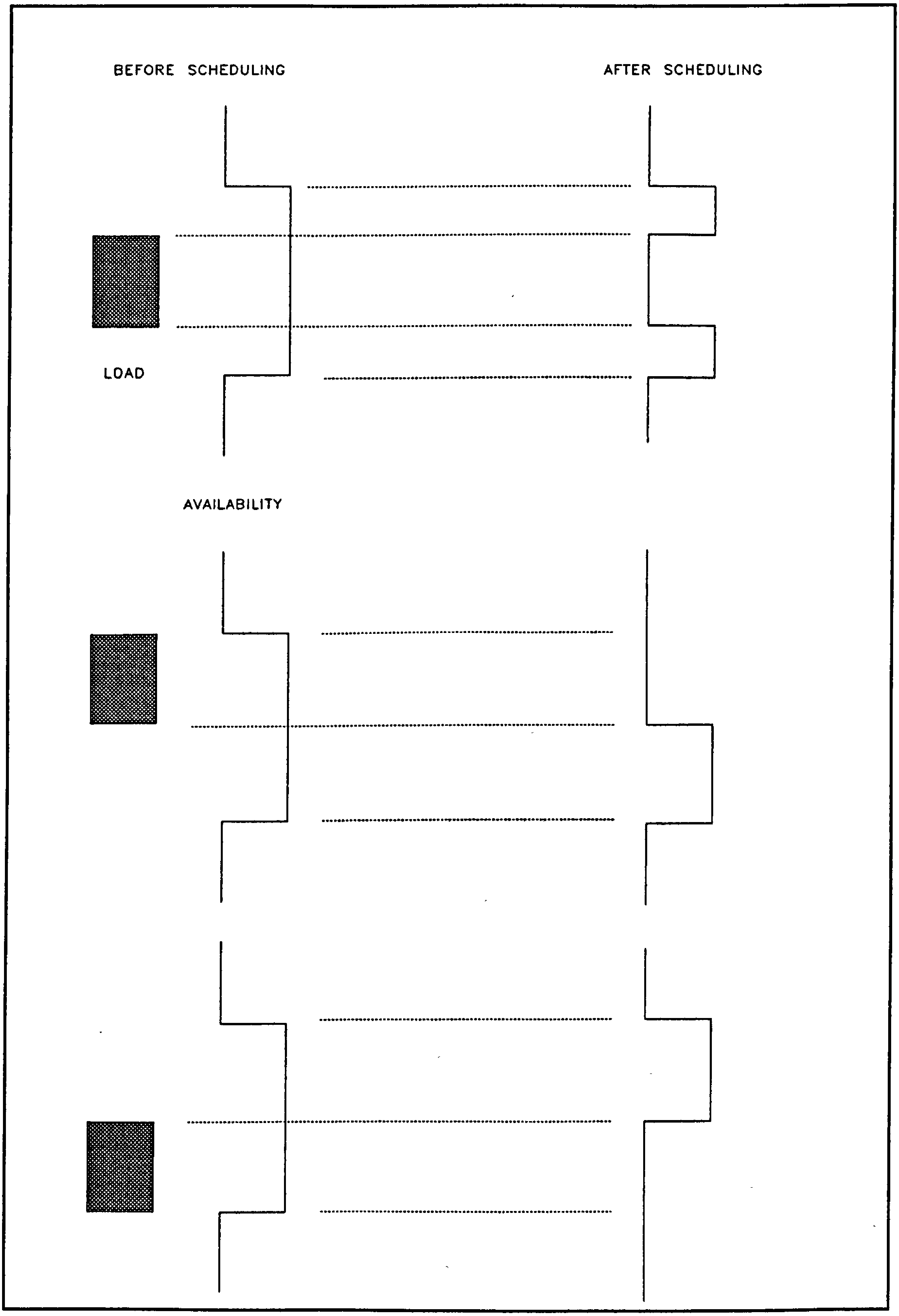


Figure 1.2: Loading Availability Slots

and a realistic short term planning tool has to have ways to cope with them.

Thereby, the discussion on finite capacity issues is understood as being a natural extension of the primary objective of this thesis.

1.3. OBJECTIVES

Many of the obstacles surrounding the applicability of CPA to production scheduling have been expressed qualitatively rather than quantitatively. Despite such arguments being based on common sense, this study decided to take the risk of, at least, proving the obvious, i.e., to investigate up to what extent is possible, if so, to apply CPA to the sequencing problem. The class of problem is not obviously the single string type job shop, but the tree structure provided by the assembly shop. Such an investigation has an intuitive appeal since assembly jobs may essentially be represented by convergent network structures. Summarizing, this study aims to investigate the feasibility of applying CPA as a sequencing priority rule for complex job structures such as assembly jobs.

An additional goal of this study refers to developing a scheduling approach by applying a CPA type sequencing rule in a finite capacity environment. Such a system does not intend to be a fully workable system, since the complexity involved in practical problems is far beyond the scope of this study, but to delineate the potentiality of adopting the proposal. Moreover, the alternative of incorporating methods such as, overlapping operations and splitting batches, into a scheduling system are also analyzed.

1.4. METHODOLOGY AND THESIS ORGANIZATION

By investigating the current literature on assembly shop sequencing rules (chapter 2) a proven and effective priority rule is selected in order to be used as a benchmark against the proposed rule. Moreover, the way capacity is approached by relevant production systems is presented.

The second step in this investigation is carried out by developing a non-backtracking priority sequencing rule for assembly shops, which uses the CPA method and forward-backward passes (chapter 3). The adoption of CPA as a sequencing rule is based on the assumptions that flow time and work-in-progress are minimized by a sequencing rule which improves the synchronization among operations and components. The generated heuristic algorithm describes the steps required by the above rule.

The proposed rule is tested against the benchmark rule in terms of the following performance measures: (i) flow time, (ii) waiting time, (iii) tardiness, (iv) lateness and (v) number of tardy jobs (chapter 4). To accomplish such a testing, an experimental assembly shop is designed to allow random simulations to test the proposed and benchmark rules on different types of BOMs.

In chapter 5, conclusions are drawn from a statistical analysis carried out on the simulation experiments just described.

The rule developed in chapter 3 and evaluated in chapters 4 and 5 is then applied to a finite capacity environment in chapter 6. In addition to the heuristic algorithm, chapter 6 considers the possibility of splitting batches as an alternative to cope with capacity restrictions.

CHAPTER 2

Literature Review

This chapter initially presents some of the relevant achievements in scheduling research. This attempt is somewhat succinct due to the dynamic and evolving research field related to industrial scheduling. Emphasis is given to assembly shop research on heuristic sequencing rules, a research subject that has become increasingly relevant over the years. Some of the studies on the relationship between a production system and the capacity issue, particularly on the finite capacity scheduling approach, are also presented. Such an aspect is coherent with the considerations introduced in the last chapter, *section 1.3.3 Capacity.*

2.1 SEQUENCING

2.1.1 JOB SHOP

The most guaranteed method to find the optimum sequence of jobs onto machines is by enumerating all alternatives and then selecting the best one according to a certain pre-defined performance criteria. As a complete enumeration of all $(n!)^m$ (n jobs and m machines) scheduling alternatives is not usually feasible, short-cut methods, have to be developed. Optimal solutions could be found for a small number of machines. However, it is impossible to check optimality for large number of machines either in flow-shop or job-shop **(Elsayed and Boucher, 1985)**. Moreover, compromising measures had to be done in order to adjust the already mentioned complexity of the production environment to the theory. For instance, the following methods assume a static job arrival pattern, deterministic processing time and no setup consideration.

(i) N jobs/1 machine: mean flow time and mean job lateness are minimized by the Shortest Processing Time (SPT) rule.

(ii) The optimal sequence of N jobs/2 machines algorithm to minimize the makespan **(Johnson, 1954)**.

(iii) N jobs/3 machines. Optimum solutions may be found, as long as all jobs are in a flow-shop pattern (no passing is allowed).

(a) The Johnson algorithm may be used as long as: the minimum processing time of all jobs on either machine 1 or machine 3 is greater then or equal to the maximum processing time of all jobs on machine 2.

(b) Another way to get the optimum sequence to minimize the makespan is through partial enumerative methods such as the branch-and-bound algorithm developed by **Ignall and Schrage (1965)**.

(iv) 2 jobs/M machines. A graphical model which presents a sequence able to minimize the makespan is presented by **Hardgrave and Nemhauser (1963)**.

The most widely used enumeration method is the already mentioned Branch and Bound approach. Dynamic Programming adopts a similar approach in the sense that both methods rely on listing and eliminating processes. Branch and Bound has a structure similar to a tree. The structure starts from an initial node at which no job has yet been scheduled. Then additional jobs are attached to the sequence at each node until the whole set of jobs is scheduled. For details on enumerative and partial enumerative methods see **Baker (1974)** and **Conway *et al.*,(1967)**. Its only limitation refers to the excessive computational time still required for large applications. An alternative in the tree search approach which reduces the computational time is provided by heuristic methods with local neighbourhood search **(Spachis and King, 1979)**.

Due to the unsuccessful attempts to create an optimizing procedure for the general job-shop problem, heuristic rules started being increasingly considered. For the N jobs/M machines, static flow-shop, two heuristics provide good solutions, though no optimality checking is available **(Campbell *et al*)**. The basic assumption of the heuristic approach, for the job-shop case, was to decompose the total system into a series of interrelated single machine scheduling problems **(King, 1976)**. Without means to guarantee an optimum solution, good alternatives could be found by applying priority rules (dispatching rules) similar to those of the N jobs/1 machine case. The most common methods to select jobs are:

(i) at random (Monte Carlo).
(ii) First-Come-First-Served basis.
(iii) according to the Earliest Due Date.
(iv) Shortest Processing Time.
(v) according to the Earliest Operation Due Date.
(vi) Slack per Remaining Operation (S/OPN).
(vii) Critical Ratio.

A complete description of job-shop sequencing rules may be obtained in **Blackstone *et al.* (1982)**.

2.1.2. CRITICAL PATH ANALYSIS AND OTHER NETWORK TECHNIQUES

The Gantt chart has been the primary technique for scheduling single job production (project production). Since the late 1950s a number of planning techniques, which are able to handle several projects simultaneously have been developed. These techniques initially define the dependence between planning stages through the activities network and then identify the critical activities path, since it will regulate the scheduling process. Finally, resources are then allocated. An interaction process aiming to evaluate the feasibility of the correspondence between planning and allocation is accomplished throughout the process. The best known examples of this approach are PERT (Program Evaluation and Review Technique), CPM (Critical Path Method) and GERT (Graphical Evaluation and Review Technique). Further details may be obtained in **Malcolm *et al.*, (1959), Kelley (1961) (Elmaghraby, 1967)** and **Awani (1983)**.

Rodammer and White (1988) sees the machine-scheduling problem as a special case of resource-constrained project scheduling. Their conceptual similarity is emphasized when both are modelled as networks. Professor Rodammer further adds that the heuristic algorithms embodied within commercial project scheduling software might well be adapted to production scheduling.

A successful adaptation of the critical path method to scheduling operations in a medium-sized tool shop is reported by **Heuser and Wynne (1963)**. The scheduling procedure consisted of translating a new job into a network, critical resource levelling and production of the schedules for the activities involved. Later, **Cooper (1972)** described the characteristics of a system called NIMMS POWER which applied networks to plan workshop scheduling. One of its interesting features was called *Individual Resource Availabilities*, which considered the availability of each resource in terms of time and size. In this sense, this system represents one of the first commercial packages of note to make use of finite scheduling.

The PYRAMID production control system resembles a manufacturing stage chart, where the time allocated to perform a particular manufacturing function is calculated on the basis of the actual (batch) processing requirements plus a queue allowance **(Buxey, 1989; and Corke, 1977)**. PERT/CPM type logic is used to advance job start times to alleviate machine overloads. If overloads still can not be solved, a job priority order is input, and forward loading to finite capacity provides a new feasible schedule.

2.1.3. ASSEMBLY SHOP SEQUENCING RULES

One of the assumptions of most job shop studies, e.g. **Blackstone et al. (1982), Panwalker and Iskander (1977)**, is that a job or customer order consists of a single component on which operations are performed. In most realistic situations, a job consists of a number of components, which will have to go through several operations, including assembly ones, to become an assembled final product at the end of the process. Moreover, multi-level assembly jobs are far more frequent than simple string type jobs as stressed by **Adam *et al.* (1987)**. Such an approach has contributed valuable insights into scheduling problems, but a number of peculiar aspects which just arise in multi-level jobs have been disregarded. The assembly shop adds a new dimension to scheduling problems, i.e., the *coordination* among parallel operations required by assembly operations. In other words, the readiness of an assembly operation depends upon the readiness of all its components. If in a job shop delays are caused by lack of sufficient productive capacity, in an assembly shop the lack of synchronization among operations performed on components, which are components of the same assembly, also contributes to increasing waiting time. Therefore, synchronization is a fundamental aspect when dealing with multi-level assembly jobs. Over the years, an increasing number of investigations have been reflecting the more realistic assembly shop environment (**Fry *et al.*, 1989; Huang, 1984; Rochette and Sadowsk, 1976; Russel and Taylor, 1985a; Scully, 1980; and Trilling, 1965**).

Maxwell (1969) developed one of the first simulation studies on assembly shop sequencing rules. In this pioneer study he tested a number of job shop rules against rules specially designed for assembly shops[2]. The tested assembly job consisted of just two levels with variation on the number of branches. *Number of Uncompleted Branches-Shortest Processing Time* (NUB-SPT), one of the assembly sequencing rules, outperformed all the others in several performance measures. The NUB-SPT rule establishes that "the priority of an operation is the total number of branches of the job for which not all the operations of the branch have been completed. This number is computed for each operation in a queue every time an operation is to be selected from the queue. (SPT is used to break ties.)" However, as **Scully (1980)** explains, NUB-SPT[3] has "a main and serious disadvantage" since "it can only be applied to job shops with only one final assembly operation. It has no meaning when sub-assemblies are involved."

---

[2] Appendix 1 presents details on several priority rules including the sequencing rules designed for assembly shop from the Maxwell study.

[3] Scully (1980) refers to NUB-SPT as NUJOB-SPT

Another contribution of Maxwell's study refers to the introduction of a new performance measure specially designed for assembly shops called *staging delay* or *staging time*. Staging time is the delay encountered by segments coming into an assembly point when they have to wait for one another before their assembly operation can start.

According to the view that the success of a rule, when applied to an assembly shop, depends upon its ability in incorporating the inherent complexity of product structure in analysis, **Scully (1980)** developed what he called *operation float* information. Operation float is defined as being job status information which can be up-dated and used for assigning a priority to an operation. The float is computed by performing a critical path analysis on the remaining operations sequence network, and putting the latest possible job completion time equal to the earliest possible completion time. Scully showed that the addition of *operation float* to *operation slack* improved the performance of priority rules involving slack[4]. To test this hypothesis Scully developed a simulation study similar to the one used by **Maxwell (1969)**, including the fact that sub-assemblies were not allowed. Scully justifies the improvement supplied by the Float approach on slack rules by the experience in using large PERT systems: when resource availability is tight, the resource should be allocated to critical activities first. Scully explain that when the operations routing is complex, as is the case in job shops with assembly operations, the operations sequence network begins to resemble the PERT-networks used for planning and controlling large projects. According to Scully, the operations sequence network can be subjected to CPM time analysis, the critical path of the network can be identified, and the float of the various operations computed.

The already mentioned NUB-SPT rule outperformed the Float based rules proposed by Scully in terms of mean flow time, mean job tardiness and percentage of jobs late. Such an occurrence did not invalidate Scully's hypothesis,

---

[4] Appendix 1 present details on several priority rules including the Slack rules and the Float rule.

but Scully considered that the NUB-SPT rule was privileged by the method used to assign due-dates, which involved the length of the critical path. **Russel and Taylor (1985a)** contradicts such an interpretation by concluding that "the method by which job due-dates are assigned does not affect the selection of labour assignment or item sequencing rule." This study aimed to evaluate scheduling policies in a dual resource constrained assembly shop. Labour was added to machines as an extra constraint. The considered policies were (i) Due Date Assignment (Total Work Content, Longest Path and Modified Longest Path, which considers the longest path multiplied by the number of branches), (ii) Labour Assignment (Longest Queue, Longest Waiting Time), (iii) Sequencing Rules and (iv) Job Structure (Tall, Flat and Mixed) affects the sequencing rules performance, as well as, the due-date assignment method and the labour assignment. **Fry *et al.*, (1989)** corroborates this research in the sense that there is a strong relationship between product structure and sequencing rule performance.

In another study, the same team (**Russel and Taylor, 1985b**) concluded that as the BOM gets taller (more levels) the probability that the job will finish late increases. They also concluded that the SPT sequencing rule improves as product structure gets taller. Interestingly however, a subsequent study suggested that the performance of the SPT rule does not improve when the product structure gets taller (**Fry *et al.*, 1989**). Moreover, they concluded that SPT is not appropriate for a shop that performs assembly operations. Nevertheless, the same study implied that SPT when combined with rules which incorporate product structure information produces good results, as demonstrated by the performance of the LVLSPT rule. This rule prioritizes operations which are located in the highest level of the BOM. Ties are broken by SPT. In this fashion, LVLSPT processes first those jobs which are closest to completion. It is interesting to note that **Rochette and Sadowski (1976)** had already suggested that sequencing rules which excel in a job shop environment are not necessarily appropriate to an assembly shop.

Adam *et al.*, (1987) accepting Maxwell's recommendation (**Maxwell, 1967**), divides the lead time into two components, the flow time and the staging time. The rationalization is that lead time reduction may be achieved through staging time reduction, which by its turn, is reduced by proper coordination of components towards assembly. According to previous research, such a coordination is achieved through rules which incorporate the structural complexity of jobs. Accordingly, a new rule called TWK-RRO was introduced. This rule paces the completion of parallel components over the entire BOM as a job progresses towards completion. This rule initially sequences items by least total work remaining (TWK), thereby establishing priorities across jobs. Items with similar TWK, i.e. components of the same job, are sequenced by the Relative Remaining Operations (RRO) rule, which paces the completion of components for each assembly, across assemblies and up through the BOM. It is interesting to note that the least TWK rule was predominantly the best rule computed by **Siegel (1971)** with respect to mean lead time. **Adam *et al.*, (1987)** performed comparisons between the mentioned NUB-SPT and the TWK-RRO, also with its variant TWK-RRP, where RRP means Relative Remaining Processing time. The performance of the proposed rules were not significantly different from NUB-SPT. The proposed rules have the advantage over NUB-SPT since they may be applied to product structures with more than one level.

Philipoom *et al.*, (1991) conducted an experiment to evaluate the performance of the so called multi-attribute based sequencing rules[5], i.e. rules which incorporate attributes of both job shop and assembly shop. Another contribution to this research was the proposal of a new set of sequencing rules called Importance Ratio (IR). The experiment consisted of eight sequencing rules on three different product structures in a hypothetical assembly shop composed of ten work centres and one assembly station. Service time at the assembly station was assumed to be zero. The rules were divided into three sets, TWK variation rules, SLACK rules and the proposed IR rules. To date, the research suggests that sequencing rules which incorporate attributes of both job-shop and assembly-

---

[5] Appendix 1 provides details on the multi-attribute based sequencing rules.

shop do not necessarily produce the best results. Moreover, multiple measures of inventory and tardiness must be considered when choosing a rule to supply a compromise solution.

The simulation results indicated that:

(i) Variations in the TWK rule, such as TWK-RRO and TWK-RRP, the rules proposed by **Adam** ***et al.*** **(1987)**, do not significantly outperform the simple TWK-FIFO rule.

(ii) Variations on the SLACK rule, such as MS-IR and MS-TWK do not offer significant improvement over slack per remaining operation rule (S/OPN).

(iii) Rules especially designed for assembly jobs, such as TWK-RRO, IR-S/OPN and IR-TWK, significantly outperform S/OPN.

(iv) **Philipoom** ***et al.*****, (1991)** conclude that "IR-TWK performs as good or better than TWK-RRO for every performance measure and significantly outperforms TWK-RRO for tall structured jobs. These results suggested that IR-TWK is more appropriate for sequencing assembly jobs than any of the other rules tested."

As the assembly sequencing rule proposed by this research will be tested against the IR-TWK rule, more details are given below. The IR-TWK algorithm utilized in the computational tests is supplied in appendix 2.

IR prioritizes an item based on the ratio of remaining number of operations on a particular branch (or path) to job completion, to the remaining number of operations on the longest path to job completion. The example utilized by **Philipoom** ***et al.*****, (1991)** is given in figure 2.1. The squares represent components and the circles represent operations. The filled circles represent operations that have been completed. " The remaining number of operations along item B's path leading to the completion of A is three (one operation for item B plus two for item A). The maximum number of remaining operations is five, along item C's path to A's completion. The importance ratio for item B4 is thus , 3/5 and for C1 is 5/5 or 1. An IR of 1 indicates that the item in its current operation is the one most likely to delay the completion of the overall job. Items with the largest IR are sequenced first. It is easy to see from the above example that the maximum IR for any operation will be 1. It is also evident that every job being processed through the shop at any point in time will have an operation whose importance ratio is equal to 1. Thus, in the event that those items with identical IRs are enqueued at the same resource, the IR rule must include some method for breaking ties in sequencing priorities." **(Philipoom *et al.*, 1991)**. In the case of the combined rule IR-TWK, ties are broken by processing first the items which have the least total work remaining. In short, IR considers product structure and staging delays by identifying the most critical operation for each job. When IR is combined with TWK, job progress is also taken into account.

2.2 PRODUCTION SYSTEM AND THE CAPACITY ISSUE

Gelders and **Wassenhove (1985)** discuss how production and inventory control systems such as MRP (Material Requirement Planning), JIT (Just in Time) and OPT (Optimized Production Technology), behave in environments where capacity constraints are prevalent. JIT is regarded as the suitable system to be used in production environments of stable demand and repetition on process. On the other hand, MRP is viewed as the system for many product options, frequent engineering changes and fluctuating product demands. In Manufacturing Resource Planning (MRP-II) systems, the Capacity Requirement Planning (CRP) module validates the production planning prepared beforehand by the MRP. The CRP loads each work centre within the pre-defined time bucket in order to check if there is sufficient capacity to meet the requirements dictated by the MRP. If necessary, the MRP may be revised or manual adaptations required to adapt the load along the production period. Therefore, it is not enough to check the capacity profile since the problem intrinsically lies in the lack of synchronisation between available time and the scheduling dates. **Blackstone (1989)** reasons that the adoption of Gantt type procedures, which do consider the timing aspect involved when relating capacity, should be a standard part of a capacity management software.

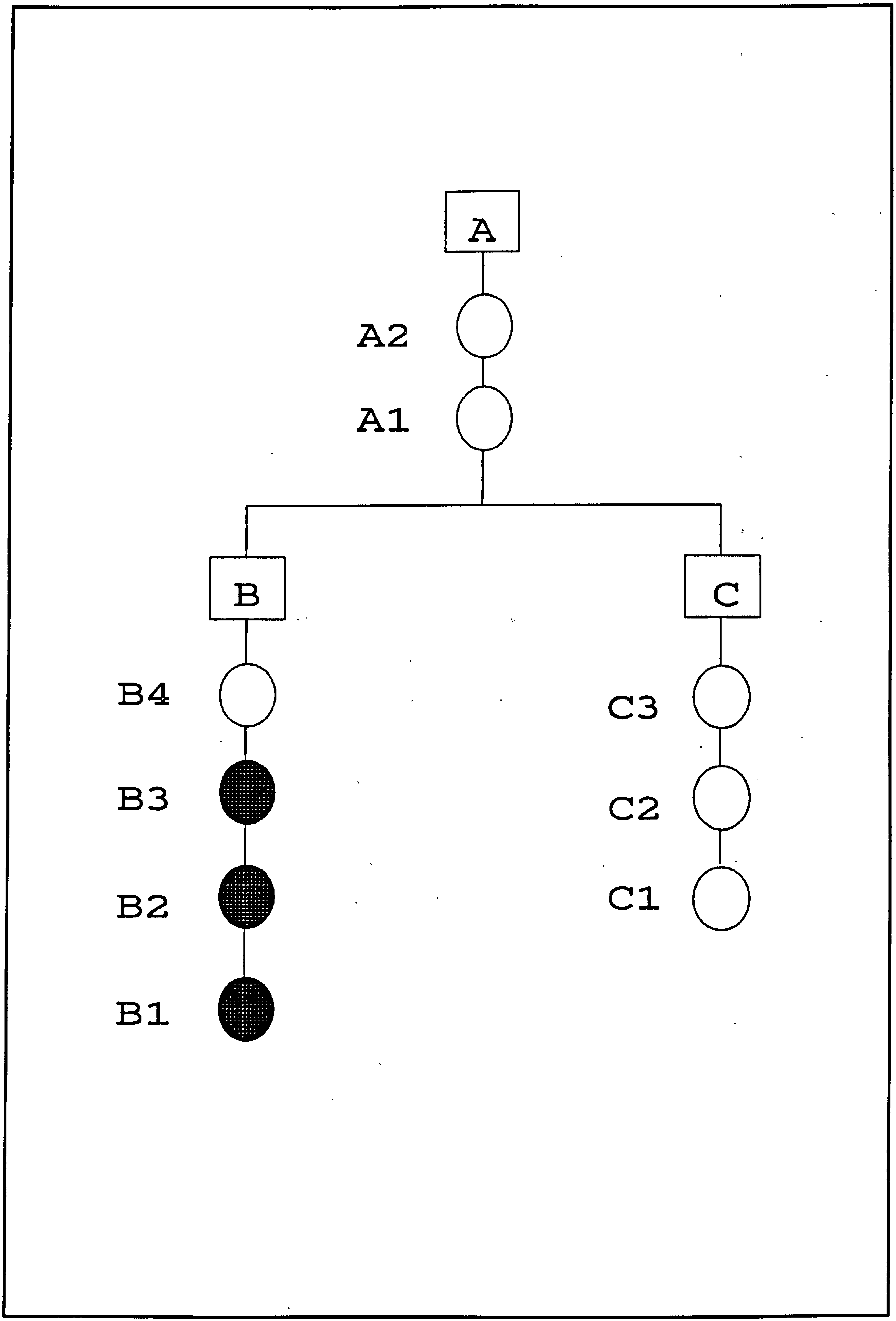


Figure 2.1: Example calculation of the Importance Ratio Rule

In terms of system applicability, **Galgut (1978)** extends such considerations by presenting a schematic representation of production control systems regarding differences among their production characteristics, as depicted in figure 2.2. Nevertheless, **Gelders and Wassenhove (1985)**, in the article mentioned above, conclude that the three systems (MRP, JIT and OPT) are not rivals, neither are they mutually exclusive. According to them, the best solution would be, perhaps, a hybrid system. OPT would come first to plan carefully the bottleneck facilities in the medium term, as a good master production scheduling tool. MRP could then be used to generate time-phased requirements, basically as a powerful information processing system for controlling thousands of items. JIT should be used in the short term and for the repetitive part of the business and to maximize throughput, a high and smooth load of work pulled through the system with minimal lead times and little work-in-progress. JIT, despite all its merit, is rather weak as a medium and long term capacity planning tool and MRP has, as one of its major pitfalls, the short term capacity planning. However, the advantages of both systems could be combined into one system. In this way, JIT would be used for repetitive items in the short term, whereas MRP would be used to balance loads in the medium term and to generate orders for non-repetitive production. An additional gain would be simulations made by MRP for months ahead, allowing if necessary, adjustments to capacity. This idea was implemented in the so called *Synchro-MRP*, developed in Japan by *Yamaha Co.* **(Hall, 1981)**. In another similar case study, **Woodgate (1989)** describes an integration of MRP technique and JIT philosophy, named MRP-III, with expert system capabilities.

The question of systems applicability has become one of the central issues in the production management literature, as demonstrated by an excellent editorial on *Finite Capacity Scheduling* by **Inglesby (1991)**. In this article, Richard T. Lilly, one of the designers of the first MRP package, regards the fact that the MRP system was originally designed for large, make-to-stock manufacturers - the only firms that could afford it. Since then, MRP systems have become too standard, when they began to be applied to any kind of manufacturing system. Defending Finite Scheduling, Lilly express his opinion by saying: "The vast majority of manufacturers have a more pressing problem than developing the material plan.

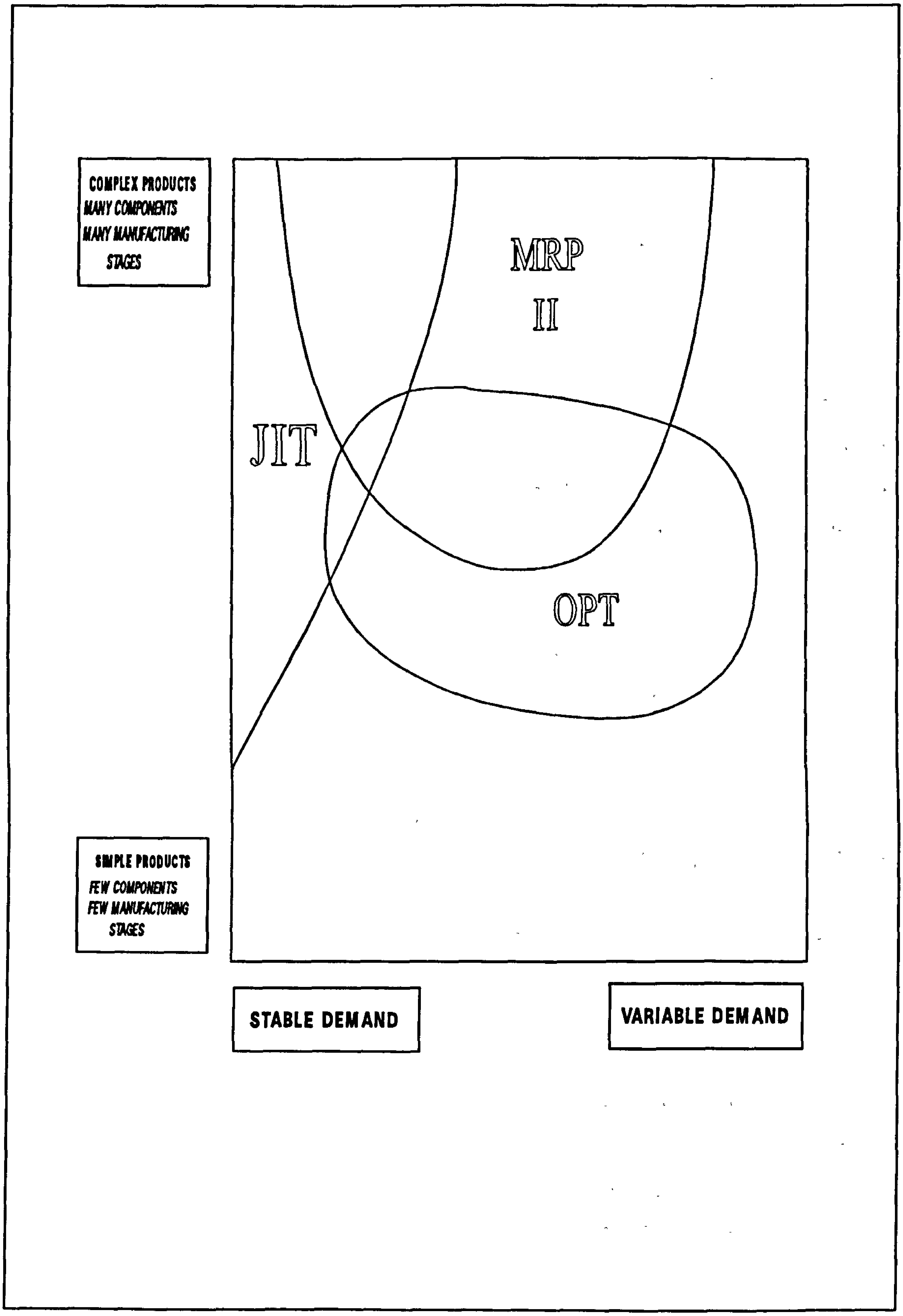


Figure 2.2: Production Systems Applicability

The problem is identifying the availability of open, non-promised capacity for a given quote, job, customer order or work order and then, based on availability, reserving the time for that job. The requirement for material is then based on the schedule of work, not on the fixed standard lead times of MRP."

Savage and **Mikurak (1985)** proposed the utilization of Finite Scheduling by MRP-II systems in order to deal with MRP's limitations on capacity issues. They define Finite Scheduling as " ... a technique to schedule and prioritize shop work within the constraints of the business." Justified by the current stage of development of computer technology, they suggest the following alternative: Instead of the classical validation process MPS/RCCP/MRP/CRP, the MRP and CRP functions should be combined into one module with the elimination of the RCCP. As an example of the feasibility of using finite scheduling in MRP, the authors present a case study of a successful integration between MRP and the OPT scheduler system.

Despite the mentioned claim that Finite Scheduling is becoming viable thanks to the advances in computer technology there is an example which comes from the 70s. The SCOPE system, developed for a *British Steel Company* plant, used infinite loading, but also loaded jobs assuming actual machine centres capacities **(Godin, 1978)**. Another system, called JOBCODE **(Hastings et. al., 1982)**, aimed to develop short term scheduling in a real time MRP environment. **Buxey (1989)**, referring to the JOBCODE system, stresses that "The heart of the system is *forward loading to actual capacity*, (normally) employing *job due date* priority and slotting in work as soon as a machine has a gap in its schedule big enough to take it. In effect, this is a deterministic simulation, with queues monitored rather than taken as average values, and whilst there is no pretension of optimality, it is virtually the only way to guarantee that a plan is feasible, even if it must sometimes deviate slightly from the original proposal. Start/finish times are generated for each operation, which constitutes a true schedule, and since the sequential loading keeps work in progress to a minimum it tends to retain its stability under actual progress conditions."

At present, the major controversies still surrounding Finite Scheduling refer to its technological momentum and its applicability. Some authors believe Finite Capacity Scheduling is to be a strong contender for the "factory of the future" **(Sarin and Das, 1988)**. Quite interestingly however, apart from OPT, a number of Finite Scheduling packages are beginning to emerge[6] **(Turner, 1991)**. **Inglesby (1991)** summarizes the controversy on Finite Scheduling applicability, by presenting the arguments of its use as a stand alone system or as an element to be linked to MRP systems. **Blackburn (1986)** concludes that "Finite Capacity Scheduling techniques, such as Kanban and OPT, are required to produce accurate production schedules."

2.3. CONCLUSIONS

*2.3.1. **Theory versus Practice**:* A job-shop environment may be understood as a number of temporary flows crossing one another in particular resources. The inherent complexity of a job-shop leads to the lack of flow visibility. Therefore, the adoption of optimizing approaches to individual sections of the shop floor, i. e., work centres or group of work centres, instead of regarding the whole shop floor is a direct consequence of such an strategy. In order to reduce the complexity of the scheduling problem, much effort of the scheduling theory has been in concentrating into portions of the problem by attempting to develop optimizing procedures, whereas commercial initiatives have been adopting compromising but broader alternatives such as heuristic methods.

2.3.2. On assembly sequencing rules

i) All rules designed so far for production scheduling in assembly shops have been scheduling operations in the forward manner, i.e., anticipating

---

[6] *Custom Manufacturing System* from ProfitKey International Inc., running in 386 PC platforms **(Sheridan, 1989)**; *Prism* from Marcam and *Finite Capacity Scheduling System* from STSC Inc., **(Turner, 1991)**; *W Squared Scheduling system* from Largotim Group Co., *Schedulex* from Numetrix Decision Sciences Inc., and *Jobshop* from Quality Manufacturing Systems **(suppliers' catalogues)**.

production by scheduling them as soon as possible.

ii) From a succession of assembly shop priority rules tested along the years, IR-TWK has presented improved results in a number of performance measures, particularly for tall structures.

2.3.3. Capacity management

i) MRP manages capacity by RCCP and then by CRP. MRP does not take capacity constraints into account when processing the requirements from the MPS.

ii) JIT has a finite loading approach through the kanban scheduling system. From the design of the production line, capacity is known by the capability of the production line itself. Such a situation is greatly facilitated by smooth flow, which in its turn, is made possible by stable demand and low variety of items/models produced.

iii) OPT has, as one of its objectives, the management of the production capacity based on the differentiation of resources between bottleneck and non-bottlenecks. The loading of raw material in the first work centre is limited by what the bottleneck resource is able to absorb (Finite loading capacity). The non-bottleneck resources are allowed to keep relative idleness in order to avoid building up of intermediate processing parts stock.

iv) Capacity requirement calculations by demanding a high volume of information to be handled, has justified capacity issues to be considered just at an aggregated level. In spite of that, there has been an increasingly number of commercial systems, which by adopting finite scheduling alternatives, have been loading operations into precise time slots.

2.3.4. On the Scheduling System: One could argue that the simplification of the production flow may come from physical alterations to the plant layout. In other words, by migrating from a functional processing type shop towards a flow-shop. According to this view, the application of Group Technology and Flexible Manufacturing Systems may take the complexity out of the scheduling task. Nevertheless, in practical situations where either the cost or period to implement such approaches may be prohibitive, new views on the scheduling procedure may also be useful. A scheduling approach which brings concepts such as flow synchronization and finite capacity to batch production is one of the basic ideas supported by this research.

CHAPTER 3

A CANDIDATE SEQUENCING RULE FOR ASSEMBLY SHOP

The *Total Work Content-Forward Backward* (**TWK-FB**) priority sequencing rule is designed to reduce, and if possible, to eliminate the staging delay component of the lead time when scheduling assembly jobs. The assumption is that improvement of synchronization of serial and parallel operations reduces waiting time as a whole, and consequently, staging delay as well. However, the production capacity available in the system is a constraining factor in terms of the resulting waiting time. The proposed rule incorporates measures to cope with capacity usage conflicts when dealing with capacity constrained manufacturing environments.

The scheduling problem related to this research is of scheduling a set of multi-level product structures. As for any scheduling procedure, the problem lies basically in sequencing operations/components/jobs in such a way as to improve one or several performance criteria. Therefore, the scheduling process is intrinsically a *selection* problem.

3.1. DEFINITION

TWK-FB may be described through three hierarchical selection levels plus a capacity checking procedure.

The selection process occurs:

(i) among product structures,
(ii) among paths, and
(iii) among operations.

Thus, the level (i) is undertaken across jobs and levels (ii) and (iii) occurs within the selected job.

The capacity checking procedure is related to constraints on capacity due to two aspects, precedence restrictions among operations and competition for limited resources.

Selection among jobs. The job itself is represented by its Bill of Resources (BoR). Like some of the best rules for assembly shops, **TWK-FB** chooses a job from the backlog according to their total work content. The one which has the *least* total work content is prioritized against the others.

Selection among Paths. Once the job is chosen, the procedure explodes the given structure into its paths. The longest path is selected to be scheduled in the *forward* manner. All the other remaining paths in the job will be scheduled in the *backward* manner, with the final operation of the critical path acting as a reference. Therefore, this part of the **TWK-FB** rule is based on the *Critical Path Analysis (CPA)*, which is largely employed in network theory.

Selection among operations. Once the path is chosen, all of its operations are scheduled through the sequence imposed by the technological sequence of operations. If the process is being undertaken in the forward manner, then operations are scheduled as soon as possible. Otherwise, operations are scheduled as late as possible. Therefore, it may be said that the selection of operations is already pre-defined by the technological sequence of operations.

The algorithm given below summarizes the *selection* process of the **TWK-FB** rule.

Step 1 Select job according to the least Total Work Content
Step 2 Select critical path
Step 3 Schedule its operations in the forward manner
step 4 Select the next longest path
step 5 Scheduled its operations in backward manner

step 6 Return to step 4 if there are still paths to be scheduled, otherwise
step 7 Return to step 1 if there are still jobs to be scheduled, otherwise,
step 8 Stop

As it may be seen, from step 2 to step 6 the algorithm describes the plain *Critical Path Analysis (CPA)* technique. However, its applicability to an assembly shop is obstructed by the dissimilarities between project and production jobs. If CPA works well for an individual network (assembly job) with no capacity constraint of any type, the same may not be said on manufacturing environments where many assembly jobs compete for limited resources. Therefore, it is necessary to develop ways to adapt CPA usage to production jobs. Such an objective is accomplished by the *Replacement Technique*, which checks any occurrence of capacity constraints throughout the scheduling process.

Initially the above algorithm is applied to an example constituted by one BOM only and unlimited resources. Afterwards, the utilization of the Replacement Technique is considered when a new BOM is added to the same example, which is then subjected to capacity restrictions.

3.2. AN EXAMPLE FOR A CAPACITY UNCONSTRAINED ENVIRONMENT

Suppose an order represented by the product structure **R** is waiting to be processed. Each operation in job **R** takes 5 units of processing time to be performed. Figure 3.1 depicts its structure and processing times. Moreover, suppose that each operation of product **R** is performed in a different resource, which guarantees no capacity restrictions either due to precedence of operations or capacity availability. Finally, the order has zero as the launch date.

Table 3.1 represents the problem in a manner convenient to the example

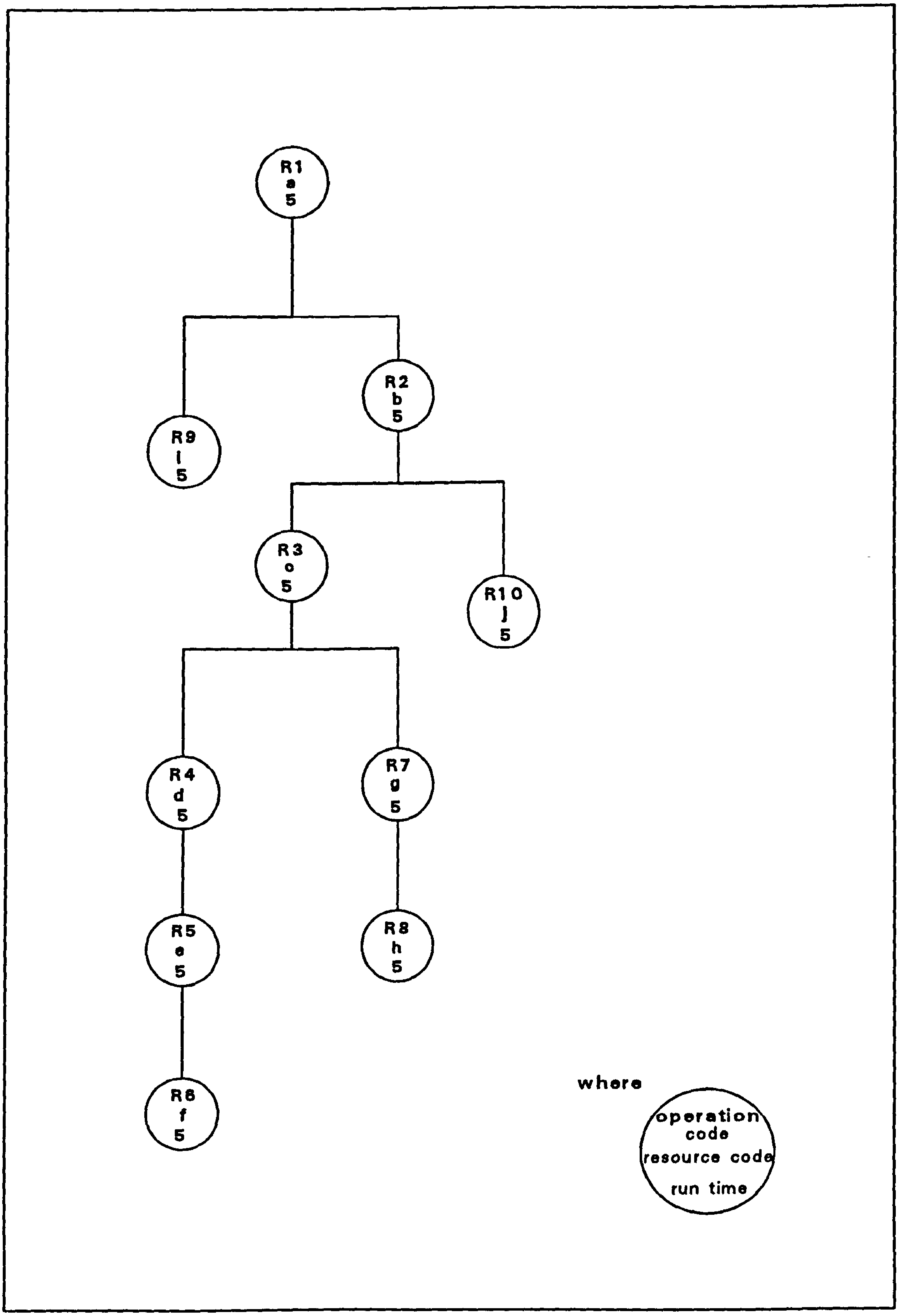


Figure 3.1: BOM of product R

purposes[7].

Table 3.1: Relaxed Case - Unconstrained Example

JOB	PATH	DURATION
R	R6 >> R5 >> R4 >> R3 >> R2 >> R1	30
	R8 >> R7 >> R3 >> R2 >> R1	25
	R10>> R2 >> R1	15
	R9 >> R1	10

step 1: Since there is just one job to be scheduled, this step is irrelevant.

step 2: The longest path **(R6>R1)** of the selected job is selected to be scheduled in the forward way. Table 3.2 illustrates the critical path schedule. The 3rd column means the starting time and the 4th column denotes the completion time of the scheduled operation.

Table 3.2: Relaxed Option, Forward Scheduling

OPERATION	MACHINE	START TIME	FINISH TIME
R6	f	0.00	5.00
R5	e	5.00	10.00
R4	d	10.00	15.00
R3	c	15.00	20.00
R2	b	20.00	25.00
R1	a	25.00	30.00

step 3: The next longest path, **R8>R1**, is scheduled in the backward manner from the already scheduled operation **R3**. Table 3.3 depicts its schedule.

---

[7]: The algebraic symbol > when applied to **R2>R1** denotes that **R2** precedes **R1**. The algebraic symbol >> when applied to **R2>>R1** denotes that **R2** directly-precedes **R1**.

Table 3.3: Relaxed Option, Backward Scheduling

OPERATION	MACHINE	START TIME	FINISH TIME
R1	a	25.00	30.00
R2	b	20.00	25.00
R3	c	15.00	20.00
R7	g	10.00	15.00
R8	h	5.00	10.00

All the remaining paths of product **R** are also scheduled in the backward manner.

Figure 3.2 illustrates the final Gantt Chart for product **R**.

From this example, it may be concluded that the single unconstrained assembly job could be scheduled in exactly the same manner as a project scheduling. In this sense, CPA when applied to assembly jobs provides the ultimate rule in terms of pacing. Actually, in the example above the waiting time among operations is zero. Furthermore, the work-in-progress is reduced since no production anticipation of any operation was allowed. The flow time is the shortest possible, but the same result could be obtained by just stacking operations according to their technological constraints. In that case, however, operations would be anticipated, which would increase WIP. Such advantages justifies the interest in analyzing the usage of the CPA approach in network problems apart from project scheduling.

One could argue however, that to focus the assembly shop in such a manner is an oversimplification of the scheduling problem. The basic obstacle for such an ideal situation refers to capacity restrictions on the manufacturing environment. In more complex production environments, CPA methods would be unfeasible. Therefore, the feasibility of CPA based rules, when applied to production scheduling, lies in additional developments.

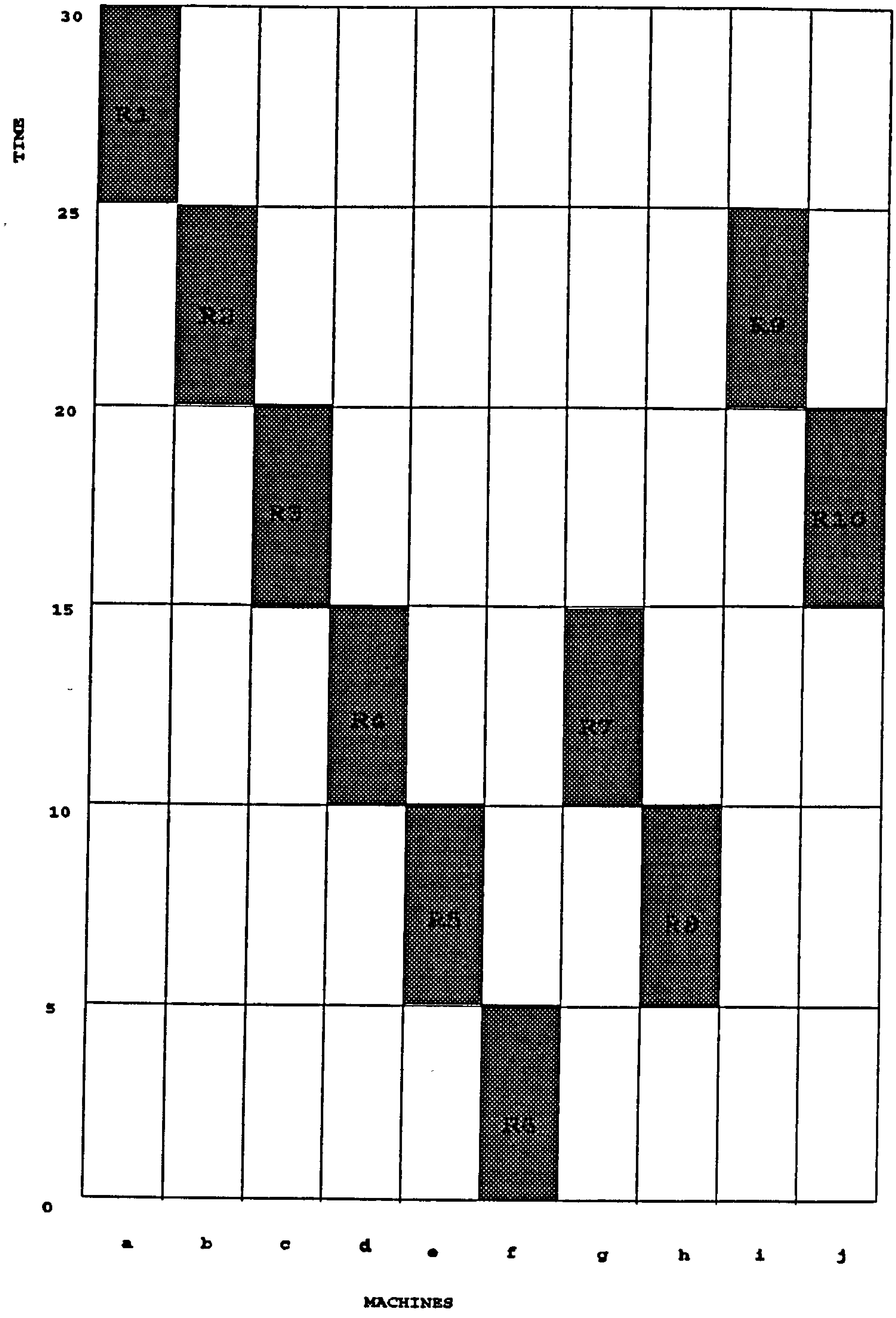


Figure 3.2: Schedule of product R represented as a Gantt Chart

3.3. THE CAPACITY CONDITIONS

Let us assume that at the beginning of the scheduling process the capacity availability of a certain resource is like one large slot. Due to the usage of capacity along the scheduling process, the total available capacity is reduced, and smaller windows of capacity availability are progressively generated. Such an occurrence becomes more crucial as more jobs compete for resources, which become more scarce as the production progresses. In such a capacity constraint shop floor, the scheduling process must consider these capacity availabilities before establishing a precise short term schedule.

The suitability of a capacity slot is considered through two basic aspects, *timing* and *size*. Requirement and availability are both analyzed in terms of size and timing.

(i) *Requirement*. Initially, the schedule of a certain operation is defined by its predecessor operation if in the forward loading, or successor operation if in the backward loading. In other words, the technological sequence of operations defines the timing of the operation being analyzed and its processing time defines the required size.

(ii) *Availability*. The capacity availability slots also have to take into account timing and size before the scheduling of an operation may take place.

Let the capacity availability be expressed by the array $A[n_k \ x \ 3]$:
where

k: the availability index in the range from 1 to n_k
n_k: the maximum number of availability slots
A_{k1}: resource code
A_{k2}: lower availability limit
A_{k3}: upper availability limit

Also, let the starting and the completion times of the operation being analyzed be expressed by W and V respectively. Where $V=p+W$ and p is the processing time.

3.3.1. FORWARD LOADING CONDITIONS

First of all, each job is sorted in an ascending manner according to their total work content TWK. Secondly, the product structure of both assembly jobs are exploded into their paths. The paths of each job are then sorted in descending way according to their durations. Table 3.1 depicted such an array.

The search for a suitable capacity slot during the forward scheduling process requires the fulfilment of the following conditions:

(i) *timing*: $A_{k3}>W$

If $A_{k2}>W$, then the operation has its starting time defined as $W=A_{k2}$, otherwise W keeps its original value.

(ii) *size*: $V-W \leq A_{k3}-A_{k2}$

During the forward loading procedure the array is sorted in an ascending manner. In this fashion, the search process starts from the most recent capacity slot. Thus, the search process is facilitated since the first slot which complies with the timing condition above is considered. If such a slot satisfies both conditions, then the slot is allocated partially or totally to the requesting operation, otherwise the searching goes on to the next slot. Once the process is done in the forward manner, it is guaranteed that there is a capacity slot suitable to be allocated in the future.

3.3.2. BACKWARD LOADING CONDITIONS

The backward search for a suitable capacity slot has to initially satisfy the following conditions:

(i) *timing*: $A_{k2} < V$

If $A_{k3}<V$, then the operation has its starting time defined as $V=A_{k3}$, otherwise V keeps its original value.

(ii) *size*: $V-W \leq A_{k3}-A_{k2}$

For backward loading the search for a suitable slot is done in a capacity array sorted in descending way. Thus, the search starts in the highest possible slot, in terms of ahead in time. Nevertheless, differently from the forward search, the backward procedure may result in being unfeasible since there is the risk of scheduling an operation to commence before the order launch date. Such an event may be due to either the non-existence of any slot or the ones which are available do not have the appropriate size. In such a circumstance, partial rescheduling of the schedule accomplished so far is required.

3.4. THE REPLACEMENT TECHNIQUE

The replacement procedure is undertaken by analyzing the availability slots during the backward search process having the timing condition as the first requisite. Along the search process, the slots are considered in terms of the ratio (θ) between the available capacity provided by the slot being analyzed and the capacity required. If a certain *capacity ratio* (θ) is equal or higher than one, then the size requirement is fully satisfied, otherwise the search process continues. The searching process finishes either if a suitable slot is found or if a launch date is reached with no suitable slot found. In that case the slot which presents the highest ratio is allocated to the operation being analyzed. Since the elected slot size is insufficient ($\theta < 1$), it is necessary to reschedule part of the operations already scheduled in order to 'make room' for the operation being analyzed.

The replacement process is undertaken in a forward manner and only considers the operations affected by the " pushing effect " caused by the insertion of the operation being analyzed. The determination of the affected operations is done by analyzing the resulting "cascade effect" from operation to operation along the routings. The replacement procedure begins by considering the superimposition between the operation under consideration and the operation that has been originally scheduled to that particular timing and resource. The superimposed operation is then rescheduled by having the operation being analyzed as a reference. As the rescheduling may cause superimposition in another operation that happens to be scheduled to the same resource, the process is repeated. In the end, all operations that have been scheduled in that particular resource are analyzed to check the existence of superimposition between operations and then rescheduled if necessary. Considering that an assembly operation happens to be rescheduled, then subsequent operations of all its routings are also checked. Such a calculation is undertaken in several paths since an assembly operation is part of several routings. Therefore, if a suitable slot is not found information on the slots are collected in order to allow a compromise choice of one of the analyzed slots.

A peculiarity of the searching process happens when $\theta = 0$, which denotes that no slot of any size whatsoever has been available for analysis. In such a case, the operation being analyzed is scheduled at the top of the array availability. As no operations have been scheduled ahead of the operation being analyzed no checking on the resource in question is requested. Therefore, the replacement procedure is limited to its subsequent operations.

3.5. AN EXAMPLE FOR A CAPACITY CONSTRAINED ENVIRONMENT

To illustrate the replacement technique consider the previous example with the following modifications:

1) The BOM of product **R** is kept, but the standard run time of operations and routings are changed as shown in figure 3.3.

2) The backlog of orders includes an additional job, which is denoted by the product **S**. Its product structure is also represented in figure 3.3.

Table 3.4 presents a schematic representation of the capacity constraint example.

Figure 3.4 sequence 1 depicts the schedule of the critical path **R6>R1** and the operation being analyzed **R7**. As **R7** directly-precedes **R3**, then there are two slots of interest in the resource **a**. The first slot is defined by the dates 10 and 15, and the second between 0 and 5, since the slot between 20 and 25 does not satisfy the timing requisite. As both slots have the same ratio 5/6, then the nearest slot, between 10 and 15, is the one allocated to the operation being

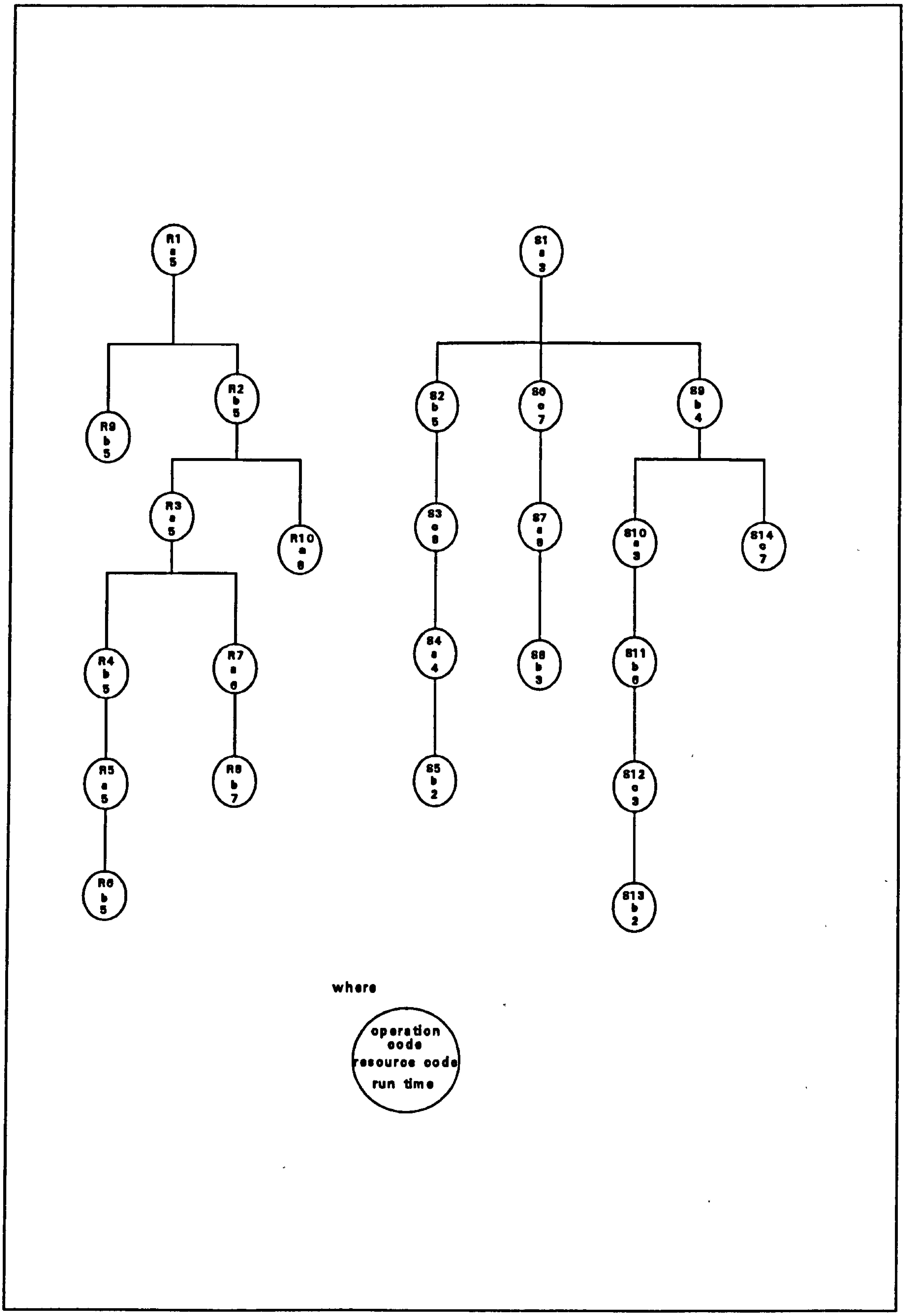


Figure 3.3: BOMs of products R and S

Table 3.4: Example of products R and S

JOB	TWK	PATH	DURATION
R	56	R6 >> R5 >> R4 >> R3 >> R2 >> R1	30
		R8 >> R7 >> R3 >> R2 >> R1	28
		R10>> R2 >> R1	18
		R9 >> R1	10
S	60	S13>> S12>> S11>> S10>> S9 >> S1	30
		S5 >> S4 >> S3 >> S2 >> S1	25
		S8 >> S7 >> S6 >> S1	20
		S14>> S9 >> S1	15

analyzed. However, the resulting superimposition between **R7** and **R3** imposes the rescheduling of **R3**, which is done in the forward manner. Such a replacement considers the new completion date of the operation **R7** as a reference. As no other operation is superimposed by the just replaced operation **R3** in the resource **a**, the procedure continues by analyzing the routing which contains **R3**. An overlap is detected between **R3** and its directly-succeeding operation **R2**, which imposes the rescheduling of **R2**. The procedure continues until all operations are checked. The procedure is similar to the insertion of the operation **R8**, which is illustrated in figure 3.4, sequences 2 and 3.

During the previous replacements just one routing was affected by the initial insertion. However, the cascade effect provoked by the insertion of the operation **R10** affects more than one routing, as shown in figure 3.4 sequence 3. Table 3.5 illustrates the backward search for a suitable slot in the resource **a**.

As it was described in section 3.3.2, the timing condition imposes that only slots that have $A_{k2}<V$ are considered when scheduling in the backward manner. Considering that **R10** directly-precedes **R2**, then $V=22$, which exclude the slots denoted by $k=1$ and $k=2$ since they are out of the range specified by the timing condition of operation being analyzed. Also, the slots denoted by $k=5$ and $k=6$ are disregarded once they are related to a resource that is different from the one

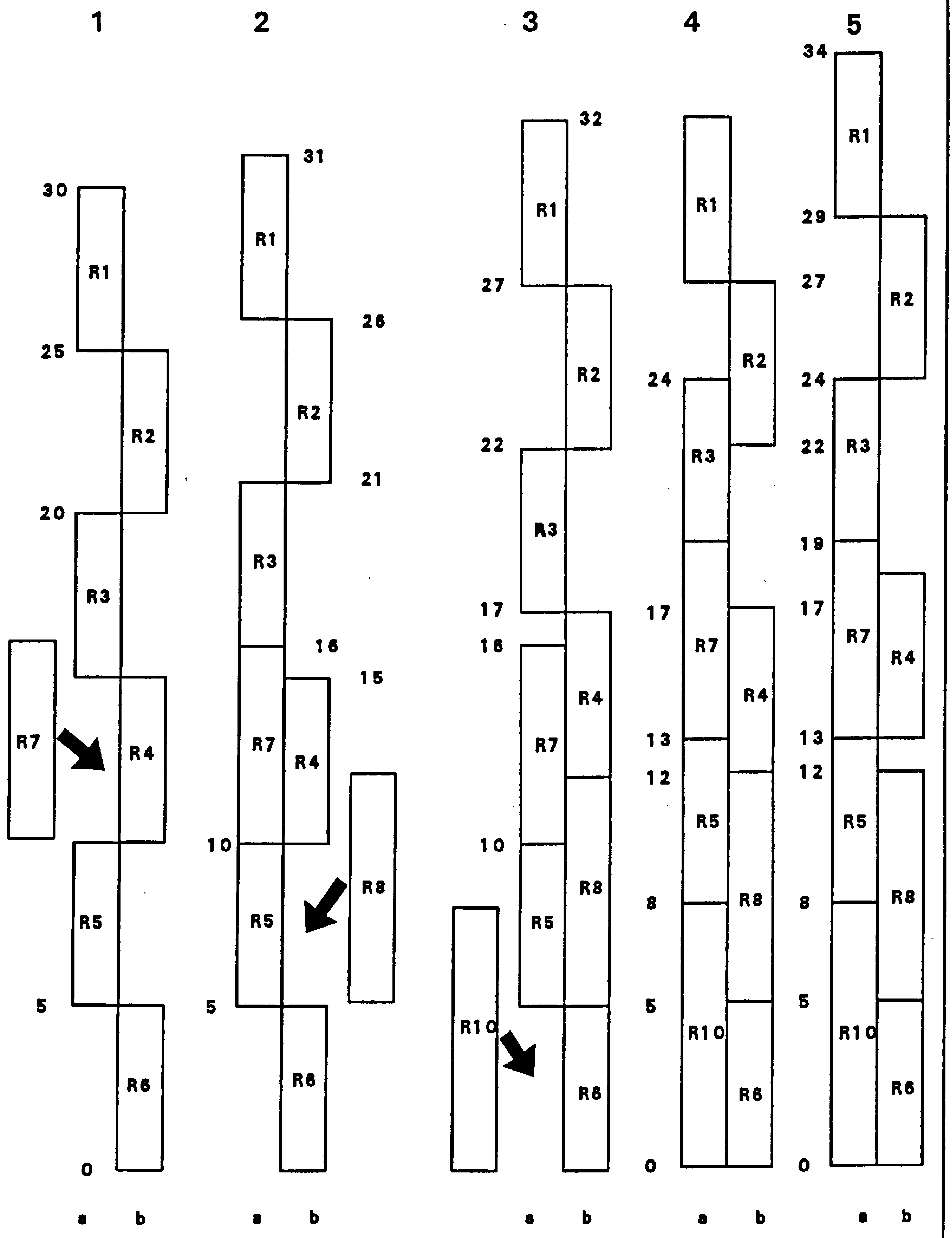


Figure 3.4: Example for the Replacement Technique

Table 3.5: Array Availability for the Constrained Example

k	1	2	3	CAPACITY RATIO θ
1	a	32	∞	NA (Not Applicable)
2	a	22	27	NA
3	a	16	17	(17-16)/6 = 0.17
4	a	0	5	5/6 = 0.83
5	b	27	∞	NA
6	b	17	22	NA

requested by the operation being analyzed.

From the eligible slots (k=3 and k=4), the last one is chosen since it presents the highest ratio. As a consequence of the insertion of **R10** in the selected slot, **R10** superposes **R5**. To overcome such superimposition, **R5** is rescheduled, which leads to the superimposition of **R5** over **R7**. Once more, the rescheduling of **R7** provokes superimposition over **R3**. The rescheduling of **R3** does not provoke any superimposition. The rescheduled operations, with exception of **R10**, are labelled in order to allow analysis on their routings, and also to prevent further rescheduling on the operations just rescheduled. Consequently, the routings **R5>R1**, **R7>R1** and **R3>R1** are checked for discrepancies resultant from such replacements. Such analysis is done by checking the operations in the segments above for the existence of overlapping[8] between operations. As a consequence, **R4** is rescheduled due to the replacing of its parent operation **R5**. In the same way, **R2** is rescheduled since it has been overlapped by its parent operation **R3**. **R2** in its turn, leads to the rescheduling of **R1**. The resulting schedule for the product **R** is illustrated in figure 3.4, from sequence 3 to 5. Table 3.6 supplies the final schedule for both products.

3.6. NOTATION

---

[8]: At this stage, *overlapping* is not incorporated into the proposal, since it is seen as an expediting measure. Chapter 6 will consider overlapping as a procedure to be incorporated in a scheduling system

Table 3.6: Schedule for the constraint capacity example

OPERATION	MACHINE	START	FINISH
R10	a	0.00	8.00
S7	a	8.00	16.00
S4	a	16.00	20.00
S10	a	21.00	24.00
R7	a	28.00	34.00
S1	a	37.00	40.00
R5	a	40.00	45.00
R3	a	50.00	55.00
R1	a	60.00	65.00
S5	b	0.00	2.00
S8	b	2.00	5.00
S13	b	8.00	10.00
R6	b	10.00	15.00
S11	b	15.00	21.00
R8	b	21.00	28.00
S9	b	28.00	32.00
S2	b	32.00	37.00
R4	b	45.00	50.00
R9	b	50.00	55.00
R2	b	55.00	60.00
S14	c	3.00	10.00
S12	c	10.00	13.00
S6	c	16.00	23.00
S3	c	23.00	31.00

It is extremely convenient for the calculation process to express the set of required information in a notation similar to the one utilized by network theory. All information related to the assembly job, such as the BOM, routings and standard times are registered in the multi-dimensional array $B[n_i \times n_p \times n_j \times 3]$.

(i) The first dimension n_i denotes the maximum number of orders (products) to be scheduled. The number of orders varies from 1 to n_i.

(ii) The second dimension n_p denotes the maximum number of paths (p) related to the job i.

(iii) The dimension n_j is related to the maximum number of elements of a path (p).

The previous dimensions refer to the 'address' of the operation. The next dimension is related to the information held in the node identified by B_{ipj}.

(iv) The last dimension has the size equal to three, since each element holds three different types of information: the operation code (B_{ipj1}), the resource in which the operation is performed (B_{ipj2}) and the standard time (B_{ipj3}).

The constraint example given in the previous section is represented by the array [B] in table 3.7. As the constraint example has two orders to be scheduled, then n_i=2. The first order (product R), denoted by i=1 has 4 paths (n_p=4), and its first (p=1) path has 6 elements (n_j). Therefore, each order (i) is related to a number of paths (n_p), where each of its paths (p) is related to a number of elements (n_j). Each element, in its turn, refers to a set of information related to the operation. As an example consider the operation **R3**. Accordingly, this operation is denoted by two nodes B_{114} and B_{123}, which are exactly the same since **R3** denotes an assembly operation which assembles two components. Thus, the second node says that **R3** is the third operation of the second path related to the first selected job. Obviously, both nodes contain the same information. The difference between them lies in their 'addresses' in terms of BOM. The information supplied by the node is as follows:

B_{1141}: operation **R3**
B_{1142}: resource a
B_{1143}: standard time equal to 5 units of time

Figure 3.5 describes the explosion of a product named **U** into its component

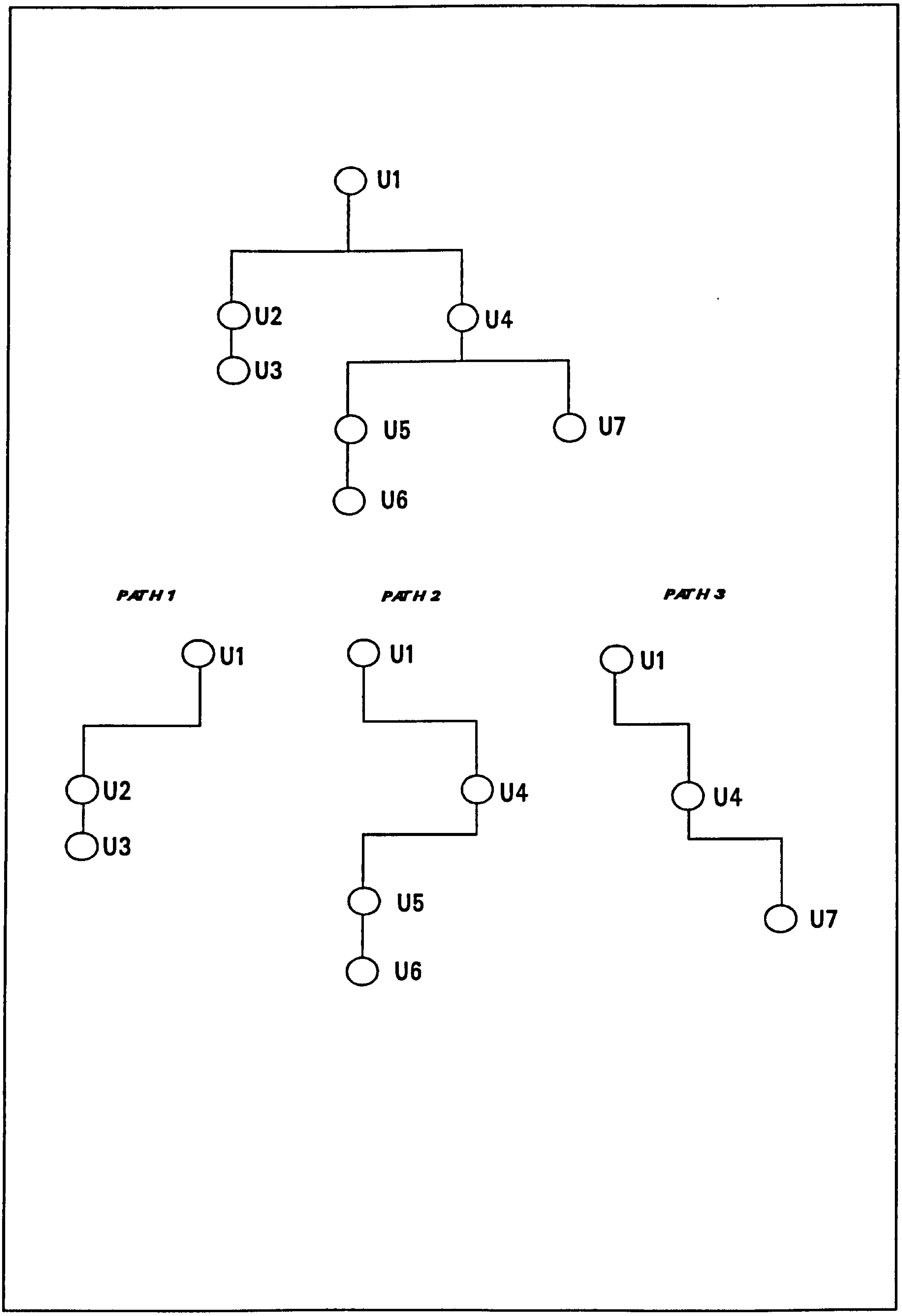


Figure 3.5: Explosion into Paths

Table 3.7: General Array [B] - A Capacity Constraint Example

i	p	j	1	2	3
1	1	1	R6	b	5.00
1	1	2	R5	a	5.00
1	1	3	R4	b	5.00
1	1	4	R3	a	5.00
1	1	5	R2	b	5.00
1	1	6	R1	a	5.00
1	2	1	R8	b	7.00
1	2	2	R7	a	6.00
1	2	3	R3	a	5.00
1	2	4	R2	b	5.00
1	2	5	R1	a	5.00
1	3	1	R10	a	8.00
1	3	2	R2	b	5.00
1	3	3	R1	a	5.00
1	4	1	R9	b	5.00
1	4	2	R1	a	5.00
2	1	1	S13	b	2.00
2	1	2	S12	c	3.00
2	1	3	S11	b	6.00
2	1	4	S10	a	3.00
2	1	5	S9	b	4.00
2	1	6	S1	a	3.00
2	2	1	S5	b	2.00
2	2	2	S4	a	4.00
2	2	3	S3	c	8.00
2	2	4	S2	b	5.00
2	2	5	S1	a	3.00
2	3	1	S8	b	3.00
2	3	2	S7	a	8.00
2	3	3	S6	c	7.00
2	3	4	S1	a	3.00
2	4	1	S14	c	7.00
2	4	2	S9	b	4.00
2	4	3	S1	a	3.00

paths. As can be seen, the node which represents the operation **U1** has three different addresses, each one of them related to a different path. In this fashion, just one array is able to represent all the complexity of an assembly job in terms of structure, routing, standard time and any other related information.

Let the Schedule be expressed by the bi-dimensional array $F[n_f \times 5]$, where:

f: the index schedule for $f \in (1, \ldots, n_f)$

n_f: the maximum number of scheduled operations

F_{f1}: operation

F_{f2}: resource

F_{f3}: the starting time

F_{f4}: the completion time

F_{f5}: register a labelled[9] operation

As an example, F_{12} denotes the resource responsible for performing the first operation scheduled.

3.7. THE REPLACEMENT ALGORITHM

In order to provide an initial comprehension on the replacement technique, figure 3.6 illustrates the situation in which such a technique is utilized by the TWK-FB rule and the effect of adopting it. Figure 3.6 also depicts the effect provoked by the benchmark rule (IR-TWK) in the same example. IR-TWK schedules operations in machines in forward manner, whatever the circumstance. TWK-FB, by making use of the replacement technique, considers availability slots even if smaller than the required capacity.

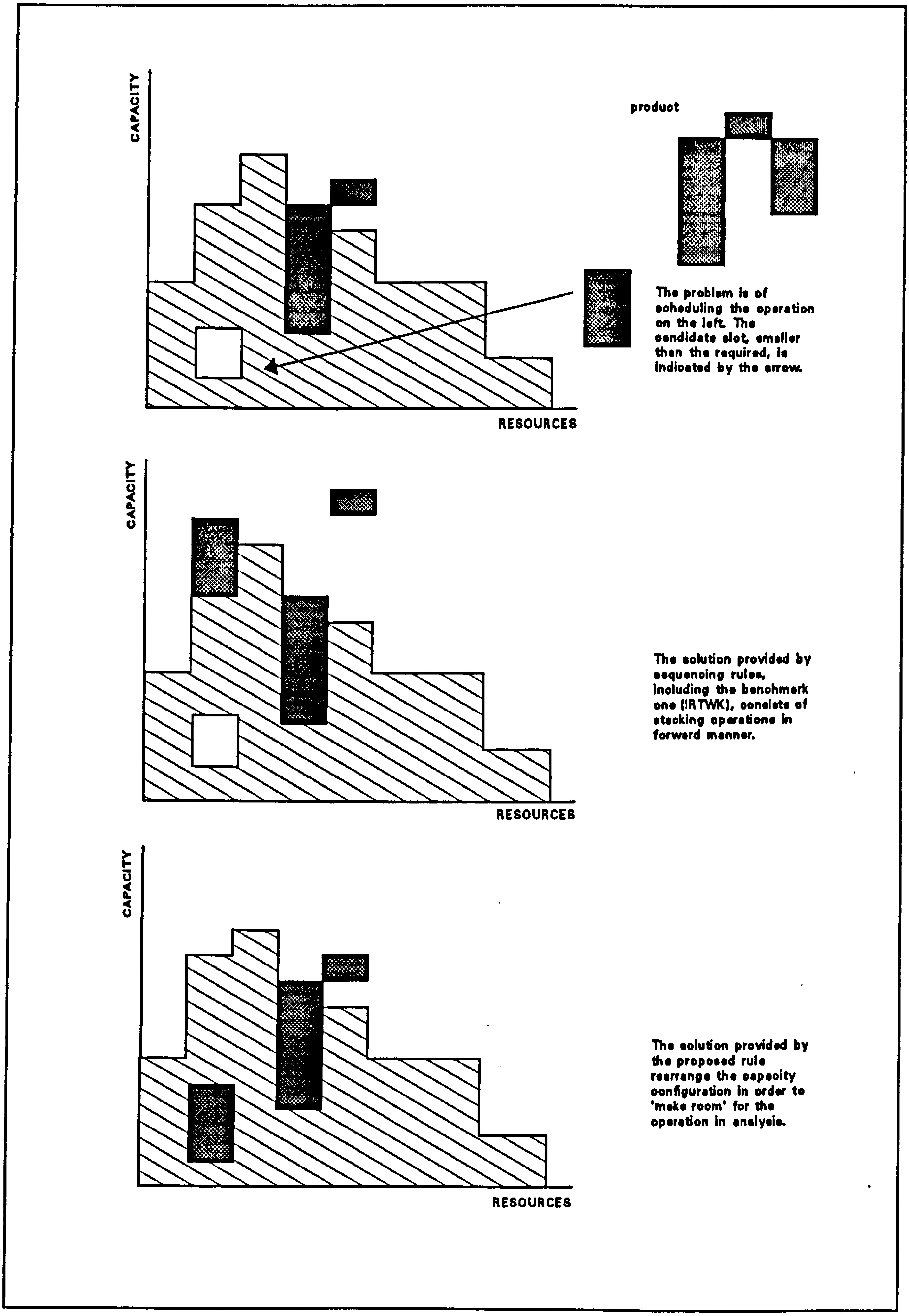
As was described above, replacement initially takes place on operations due to the insertion of an operation in a capacity slot with insufficient size. Therefore, the detailed description of the replacement algorithm starts from an operation that has been scheduled to a slot which presented a ratio lower than one.

Initially, however, two kind of conceptual differentiations has been adopted:

(i) The references V and W, are similar to $\overline{V}$ and $\overline{W}$, respectively. Where the references $\overline{W}$ and $\overline{V}$, namely upper testing limit and lower testing limit

---

[9] It will be explained later the circumstances in which an operation recieves a flag.



The problem is of scheduling the operation on the left. The candidate slot, smaller than the required, is indicated by the arrow.

The solution provided by sequencing rules, including the benchmark one (IRTWK), consists of stacking operations in forward manner.

The solution provided by the proposed rule rearrange the capacity configuration in order to 'make room' for the operation in analysis.

Figure 3.6: Comparing the Priority Rules TWK-FB and IR-TWK

respectively, refer to a *rescheduled* operation, whereas W and v refer to a *scheduled* operation.

(ii) *Superimposition* happens between operations that share the same resource. *Overlapping* occurs between operations that are elements of the same routing, but do not share the same resource.

Figure 3.7 illustrates the above considerations.

step 1. The algorithm starts by scheduling the operation being analyzed.

$$\begin{aligned} F_{f1} &= B_{ipj1} \\ F_{f2} &= B_{ipj2} \\ F_{f3} &= W \\ F_{f4} &= W+B_{ipj3} \\ F_{f5} &= LABEL \end{aligned}$$

Set $\begin{matrix} \overline{W} = F_{f3} \\ \overline{V} = F_{f4} \end{matrix}$.

step 2. This step acts within the resource being analyzed B_{ipj2}. The objective is to detect all operations which have been affected by the 'pushing effect' caused by the inserted operation F_{f1}. Set f to one. An operation is replaced if the following four conditions take place: $F_{f2} = B_{ipj2}$ and $F_{f3} \leq \overline{V}$ and $F_{f4} > \overline{W}$ and $F_{f5} \neq LABEL$. The first condition refers to the resource in which the operation has been scheduled. The next two conditions are related to the superimposition effect. Finally, the last condition avoids the rescheduling of an operation which has been already replaced in the current analysis. Figure 3.6 describes the superimposition effect. Initially, the processing time is registered as $\Gamma = F_{f4}-F_{f3}$ and then the rescheduling is given by:

$$\begin{aligned} F_{f3} &= \overline{W} \\ F_{f4} &= \overline{V} + \Gamma \end{aligned}$$

This step is repeated for all operations that share the same resource.

step 3. As the availability slots configuration has been changed due to the scheduling occurred at step 1 and possible rescheduling at step 2, it is

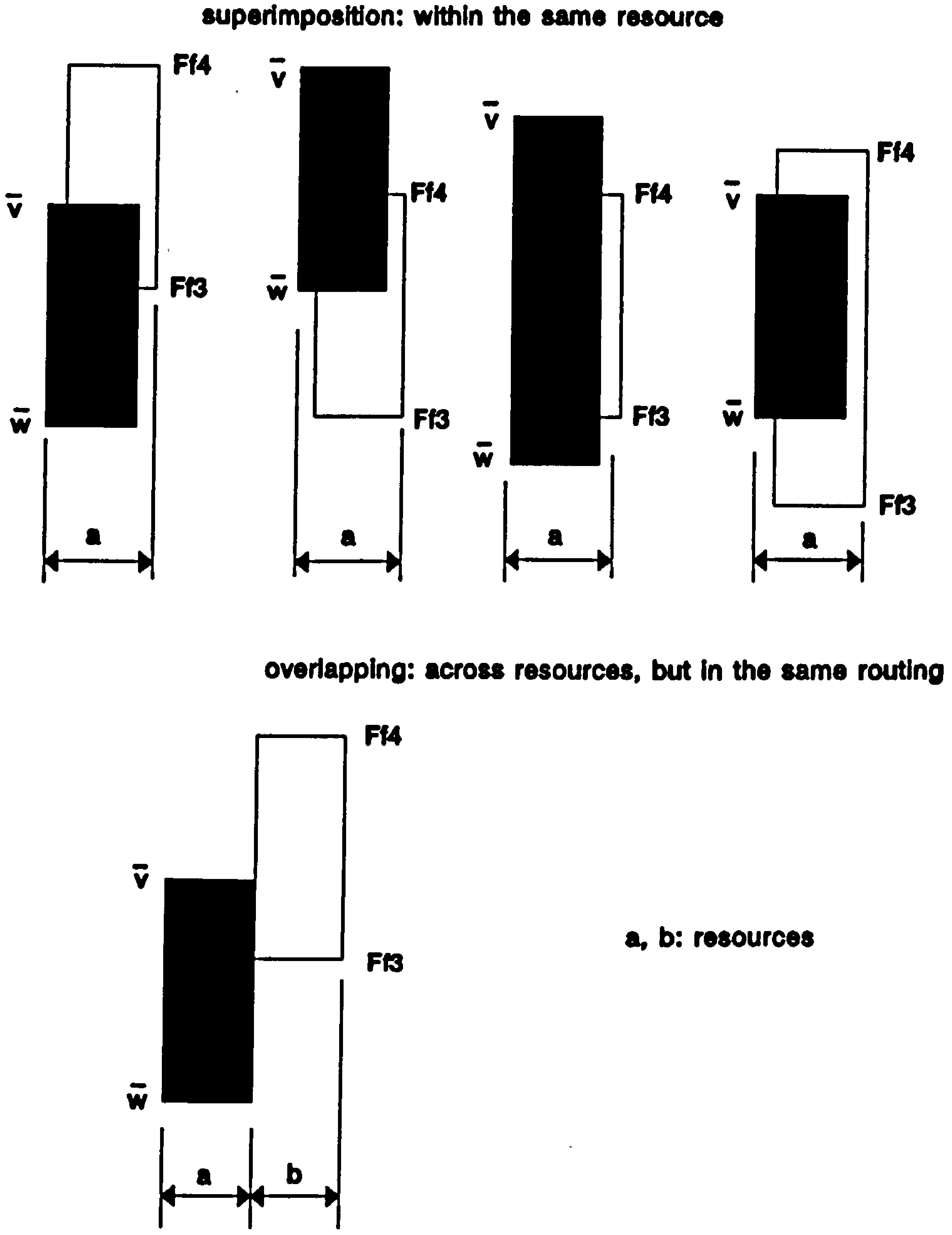


Figure 3.7: Superimposition and Overlapping

necessary to update the array [A]. The algorithm for updating the array availability is given in appendix 3.

Steps 1 to 3 resulted directly from the insertion of the operation being analyzed. The next steps result from operations labelled in the last steps and yet to be considered for possible replacement. Considering that the last steps were performed within one routing only, then the next steps may be related to several routings.

step 4. This step starts the searching process for a labelled operation. A labelled operation acts as a reference to analyze overlapping occurrences in its routing. Set f to one and go to the next step.

step 5. A labelled operation is found if $F_{f5}=LABEL$. Thus, set $\overline{W} = F_{f3}$ and $\overline{V} = F_{f4}$ and go to step 6. Otherwise, increase f by one and resume this step. When all labelled operations have been checked, clear all scheduled operation from the flag *LABEL* and finish the procedure.

The following steps, from 6 to 8, searches for the first node similar to the labelled operation. Such a procedure is necessary since the node is the starting element within the routing. The searching process is done through paths p and their elements j. There is no need of determining the job i since it has been already defined when the replacement procedure started.

step 6. Set p to one and go to the next step.

step 7. Set j to one and go to the next step.

step 8. The node has been found if $B_{ipj1} = F_{f1}$. Thus, go to step 9. Otherwise, increase j by one and resume this step. If any node in the current path does not match the condition above increase p by one and return to step 7.

step 9. This step starts the search for a scheduled operation (F_{f1}) similar to the operation that succeeds ($B_{ip,j+1,1}$) the labelled operation similar to operation B_{ipj1}. Set f to one and go to the next step.

step 10. An overlapping has taken place if $F_{f1}=B_{ip,j+1,1}$ and $F_{f3}<\overline{V}$ and $F_{f5}\neq LABEL$. The first condition locates the similar operation, the second term denotes overlapping between operations and the last condition prevents replacement of an already rescheduled operation, as well as labels the operation for subsequent routing analysis. Figure 3.6 describes the overlapping effect. Γ registers the processing time before the replacement as $\Gamma = F_{f4}-F_{f3}$.

$$\begin{aligned} F_{f3} &= \overline{W} \\ F_{f4} &= \overline{W}+\Gamma \\ F_{f5} &= LABEL \end{aligned}$$

The reference values are updated as $\overline{W} = F_{f3}$ and $\overline{V} = F_{f4}$, and the resource is registered as $\Psi = F_{f2}$. Update the array capacity and go to step 11. If the conditions above are not satisfied increase f by one and resume this step.

step 11. This step starts the search process for superimposed operations in the resource Ψ. Set f to one and go to the next step.

step 12. If $F_{f2} = \Psi$ and $F_{f3} \leq \overline{V}$ and $F_{f4}>\overline{W}$ and $F_{f5}\neq LABEL$, then the operation is rescheduled. Initially, the processing time is registered before the replacement as $\Gamma = F_{f4}-F_{f3}$.

$$\begin{aligned} F_{f3} &= \overline{W} \\ F_{f4} &= \overline{W}+\Gamma \\ F_{f5} &= LABEL \end{aligned}$$

The reference values are updated as $\overline{W} = F_{f3}$ and $\overline{V} = F_{f4}$. Update the array [A] related to the resource Ψ, increase f by one and resume this step. In this fashion all operations which comprise with the condition above are checked. When all operations have been analyzed return to step 5.

3.8 SUMMARIZING THE REPLACEMENT TECHNIQUE

In short, the replacement technique complies with the following pattern:

(i) If the elected slot presents a ratio lower than 1, but higher than zero, then the schedule of the operation being analyzed will provoke superimposition in the related resource.

(ii) An operation is rescheduled if a superimposition happens between it and a scheduled operation that shares the same resource.

(iii) All operations that share the same resource of the rescheduled operation are analyzed in order to check the existence of superimposition. The replaced operations are labelled in order to allow a subsequent checking on their routings.

iv) The two previous steps are repeated while there are operations to be checked in that particular resource.

(v) Analyze the next operation labelled for the existence of overlapping between the labelled operation and its directly-succeeding operation. If a superimposition has taken place resume the second step, otherwise resume this step while there are labelled operation to be checked.

(vi) The procedure stops when all operations are checked.

For the particular case of ratio $\theta = 0$, when no slot is eligible, the procedure is similar. The only difference refers to the reference value w, which receives the highest possible lower availability limit.

CHAPTER

4

The Simulation Model

In this chapter, the operation of a hypothetical assembly shop is designed in order to test the performance of the proposed rule (TWK-FB) against the benchmark rule (IR-TWK). It is also given the conditions in which the experiments are accomplished, the method for assigning the due-date, and the performance measures. Appendix 5 describes the simulation model.

4.1. THE EXPERIMENTAL ASSEMBLY SHOP

The hypothetical multi-stage assembly shop modelled in this study encompassed 6 single machine work centres. Routings were defined through an integer uniform distribution as in **Goodwin and Weeks (1986)**. Therefore, each operation, even the assembly ones, had the same chance of being performed by any machine. The operations required by each component were pre-defined according to the BOMs depicted in appendix 4. Operation processing times were uniformly distributed from 0.20 to 0.80 units of time. A random number generator was developed according to the *congruential method* as demonstrated in **Wu (1992)**.

4.2. EXPERIMENTAL DESIGN

A group of four experiments was conducted. Each group was composed of a specific number of jobs representing a particular type of BOM. Each group of experiment was categorized according to the product structure type. This is because previous studies have been unanimous in stating the strong influence of the BOM on the sequencing rules performance. Appendix 4 illustrates the BOMs utilized in this study.

The classification of the groups as depicted in table 4.1 aimed to represent the

Table 4.1. Designing the Experiment

EXPERIMENT No.	BOM TYPE	GROUP	No. of JOBS	PRODUCT SET	DUE-DATE FACTOR (k)	LOAD RATIO (α)
1	FLAT	1	1	A	1.2	1.00
		2	1	B	1.2	1.67
		3	1	G	1.5	3.67
		4	1	L	1.2	1.67
		5	1	M	1.2	2.67
		6	2	AB	1.5	2.67
		7	3	GLM	1.5	8.00
		8	5	ABGLM	2.5	10.67
2	TALL	1	1	C	1.2	1.33
		2	1	D	1.2	2.00
		3	1	H	1.2	3.67
		4	1	J	1.2	1.67
		5	1	K	1.2	2.67
		6	2	CD	1.5	3.33
		7	3	HJK	1.5	8.00
		8	5	CDHJK	2.5	11.33
3	MIXED	1	3	EFP	2.0	12.33
4	ALL	1	14	ABCDE FGHJK LMNP	4.0	34.33

relationship between operations and machines according to their specific BOM types. Such relationship was expressed as a load ratio (α). As the assembly shop contained 6 machines, the ratio always had 6 as its denominator. Such a ratio represented satisfactorily the environment complexity since routings and processing times were defined as random variables. Each experiment aimed to satisfy the following goals:

i) The first experiment aimed analyzing the behaviour of the sequencing rules for flat product structures in environments with different levels of complexity. As an example, group 1 represented 6 operations for 6 machines implying a load ratio of 1, and group 6 contained 16 operations altogether being served by 6 machines denoting a load ratio of 2.67.

ii) The second experiment is similar to experiment number 1, but the BOM type related is of the tall type.

iii) The third experiment was similar to the previous ones, but now testing a mixed BOM product group

iv) The fourth experiment involved all product structures considered so far plus others not considered before. The aim was to check the behaviour of the proposed rule in an extremely loaded environment.

All simulation experiments tested each product group in 10 replications of 100 observations each. Considering 18 groups altogether, the sequencing rules were tested in 36000 runs, i.e., each rule was tested a thousand times for each group.

As it was not the objective of this research to study the effect of common parts between BOMs, no commonality of parts was allowed.

4.3. ASSIGNING THE DUE-DATE

Much has been undertaken recently in terms of assignment due-dates and its influence to estimate lead times for MRP systems **(Adam et al., 1987; Goodwin and Goodwin, 1982; Goodwin and Weeks, 1986; Russel and Taylor, 1985a).** For the objectives of the present study each job was assigned a due-date under the Total Work Content Critical Path (TWKCP) allowance method, in which the length of the critical path is the base for assignment of the planned due-date. TWKCP is defined as:

$$d_i = k * TWKCP$$

where d_i is the due-date of the job i, k is an allowance parameter and TWKCP is the sum of the operation times on the BOM critical path. The second column of table 4.1. presents the allowance parameter for each group. If such an allowance was set too loose, then all jobs in both rules would finish on time. If too tight, e.g. equal to 1, all jobs would probably finish late. In both cases, no conclusion would be drawn. In order to allow suitable due-date estimations the program was run a few times with varying values of **k**. The chosen parameters were those which allowed the percentage of tardy jobs to be higher than zero and lower than 100%.

The TWKCP method of assigning due-date was used in **Maxwell and Mehra (1968)**, **Russel and Taylor (1985a)** and **Miller and Maxwell (1975)**. **Russel and Taylor (1985b)** considered it as the appropriate due-date assignment rule for an assembly shops. **Fry et al., (1989)** justify the use of TWKCP by stating that it provides a better estimation of the total processing time due to the possibility of simultaneous processing of parallel components, which was initially recognized by **Orlicky (1975)**.

4.4. THE PERFORMANCE MEASURES

The measures adopted in this study attempted to reflect different characteristics of a scheduling rule, namely makespan, work-in-progress, job tardiness, job lateness and number of tardy jobs.

4.4.1. MEAN FLOW TIME $\bar{F}$

The performance measure related to completion time was the mean flow time, which has been largely used to evaluate priority rules performance. The equation given below describes the mean flow time.

$$\bar{F} = \frac{\sum_{i=1}^{n_i} (F_i - r_i)}{n_i}$$

where,

F_i: the completion time related to job i

r_i: the launch date. As this study assumes a static arrival, the launch date is equal to zero for all jobs in the queue.

n_i: number of jobs waiting to be scheduled.

4.4.2. MEAN WAITING TIME $\bar{W}$

According to **Conway *et. al.* (1967)**, there are two basic ways to measure the work-in-progress. The first method is just counting the jobs waiting to be processed within a certain period. This is the conventional way adopted by queueing theory. The other way is to compute the amount of "work content" in a queue by summing the processing time of the jobs that are waiting to be performed. Due to the multiplicity of assembly this study adopted the second method, where the mean waiting time is used as a measure to express the work-in-progress present in the system. In Wu, (1990) this measurement was defined as:

$$\overline{W} = \frac{\sum_{i=1}^{n_i} W_I}{n_i}$$

where,

W_i: the waiting time related to job i

In order to reflect the complexity of assembly shop structures this study decided to alter the above equation as follows:

$$\overline{W} = \frac{\sum_{i=1}^{n_i} \sum_{p=1}^{n_{p_i}} W_{ip}}{\sum_{i=1}^{n_i} n_{p_i}}$$

where,

W_{ip}: waiting time on routing p (path) related to job i

n_i: maximum number of jobs

n_{p_i}: maximum number of paths on job i

As an example consider the schedule given in table 3.6 (page 43). The waiting time is computed between operations in a path, by subtracting the starting time of the successor operation from the completion time of its preceding one. The total waiting time is calculated by summing all waiting times of all paths of each order involved. Table 4.2 depicts such a calculation procedure.

Accordingly the mean waiting time is given as:

$$\overline{W} = \frac{93+64}{4+4} = 19.6\,(\mathit{units\ of\ time})$$

Table 4.2: Waiting Time calculation

PATHS OF PRODUCT R

R6 >> R5 >> R4 >> R3 >> R2 >> R1
(40-15)* + (45-45) + (50-50) + (60-60) = 25
R8 >> R7 >> R3 > R1
(28-28) + (50-34) + 0** = 16
R10 >> R2 >> R1
(55 - 8) + 0 = 16
R9 >> R1
(60-55) = 5

93

PATHS OF PRODUCT S

S5 >> S4 >> S3 >> S2 >> S1
(16-2) + (23-20) + (32-31) + 0 = 18
S8 >> S7 >> S6 >> S1
(8-5) + (16-16) + (37-23) = 17
S13 >> S12 >> S11 >> S10 >> S9 >> S1
(10-10)+(15-13)+(21-21)+(28-24)+(37-32) = 11
S14 >> S9 >> S1
(28-10)+ (37-32) = 18

64

* where 40 denotes the starting time of R5 and 15 the completion time of operation R6
** the calculation from R3 to R1 is not considered since it has been already considered in the path **R6 > R1.**

The author considers this method as being more realistic for evaluating static scheduling processes for assembly shops. This is because such a method reflects the inherent complexity of the product structure by analyzing the interelationship among operations. Moreover, the desegregation of the job into its routing agrees with what is normally done in practice, where when an assembly job arrives for processing it does not mean that all its components are stacked in line waiting for processing. Instead, just one or some of its components are initially present. The components which are not immediately necessary are kept in stock till they are requested. In order to elucidate the above reasoning consider two situations: one initiates the production of all components simultaneously at the date r_i; the other keeps in stock what is not immediately necessary. Figure 4.1 depicts both cases.

The total work-in-progress for the first case is given by the following equation:

$$Q_1 = Q_{ABC}+Q_{ADF}+Q_{BDFC}+Q_{BCE}$$

For second case it is expressed by:

$$Q_2 = Q_{ABC}+Q_{BDC}+Q_{BCE}$$

The first case has a higher work-in-progress due to the anticipation of the component c. Such an increase is expressed by the square BDFC, which might instead be represented by the waiting time FC. Therefore the computation of the waiting time along the routing just starts when materials arrive at the production line to be performed by the first operation.

The advantage of using this approach helps to differentiate situations that anticipate production (as soon as possible) from the ones which just start producing when necessary. Furthermore, this method of expressing work-in-progress not only explicitly considers the already mentioned *staging delay* across components, but also the waiting time which occurs within the same component, i.e., along serial operations.

The next three performance measures are related to due-date and largely used in a number of research, e.g., **Fry *et. al.*, (1989), Philipoom *et. al.*, (1991)** and **Russel and Taylor (1985a).**

4.4.3. MEAN ABSOLUTE LATENESS $\bar{L}$

The average deviation of job completion times from job due-dates is given by the following equation.

$$\bar{L} = \frac{\sum_{i=1}^{n_i} |L_i|}{n_i}$$

$$L_i = F_i - d_i$$

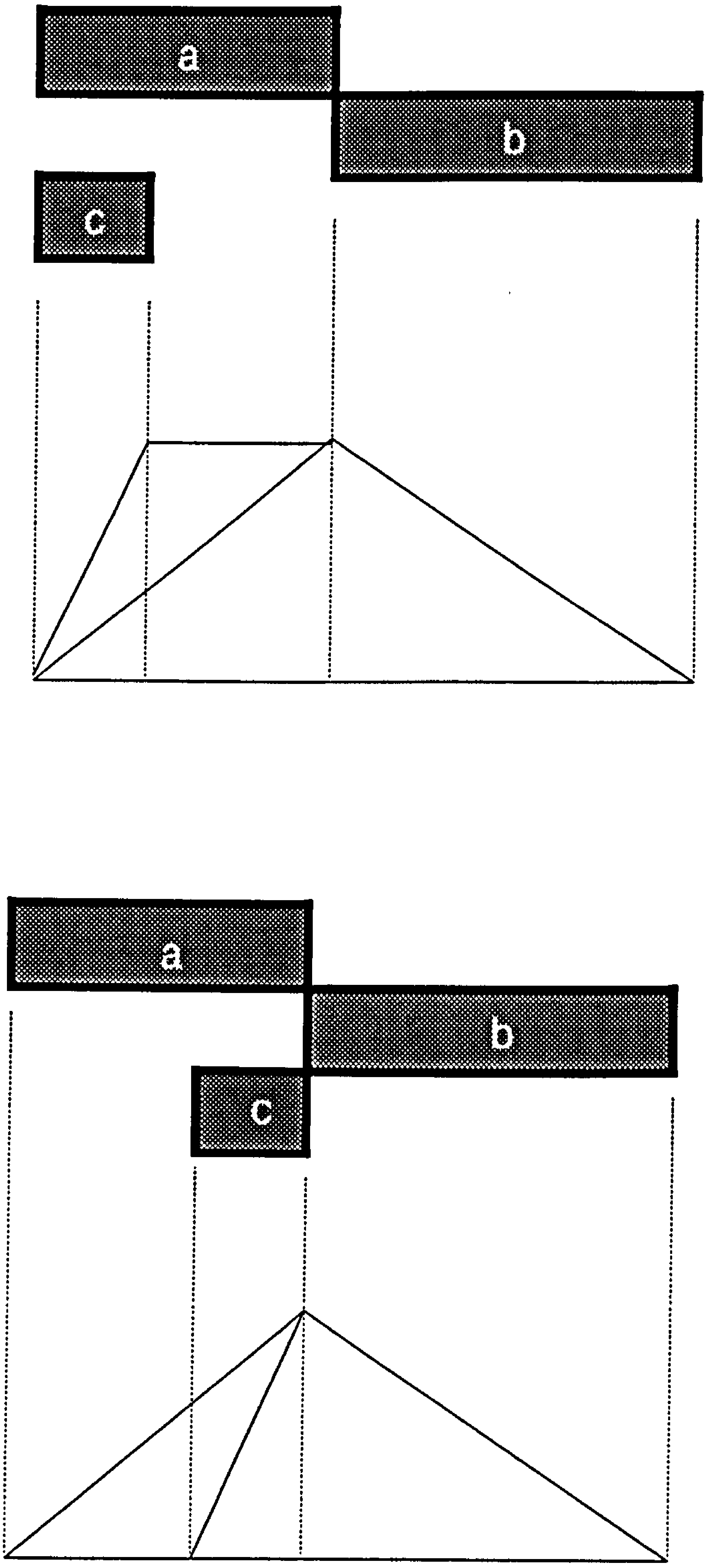


Figure 4.1: Work-in-progress and waiting time

where,

L_i: lateness of job i
d_i: due-date of job i

4.4.4. MEAN TARDINESS $\overline{T}$

This measure regards the average tardiness of all jobs completed. The equation below describes the mean tardiness.

$$\overline{T} = \frac{\sum_{i=1}^{n_i} \max(L_i, 0)}{n_i}$$

4.4.5. PERCENTAGE OF TARDY JOBS $\%T$

This measure, also related to the due-date, considers the percentage of jobs completed after their due-date.

$$\%T = \frac{\sum_{i=1}^{n_i} H_i}{n_i} * 100$$

where,

$H_i = \begin{cases} 1 & if\ L_i > 0 \\ 0 & if\ L_i \leq 0 \end{cases}$; number of tardy jobs

CHAPTER 5

Results and Analysis

5.1. EXPERIMENTAL RESULTS

Appendix 6 presents the simulation results of the experiments designed in the previous chapter. Each observation of the samples of both rules, the proposed and the benchmark rules, were generated by the same set of random variables. The statistical test used is the *student distribution (t)*, concerning two random samples. Variances of the populations are not known and not equal. The hypothesis are:

$H_0 : U_1 = U_2$; both rules present the same performance on the basis of 5% level of significance.

The opposite hypothesis is given by:

$H_1 : U_1 \neq U_2$

H_0 is rejected and H_1 is accepted if $|t^1{}_f| > t_c$, where $t_c = t(f;\zeta)$ and ζ is the level of significance **(Hogg and Ledolter, 1987)**. This study adopts a significance level equal to 5%.

The test model is given by

$$t^1{}_f = \frac{\overline{X_1} - \overline{X_2}}{\sqrt{\left(\frac{S_1{}^2}{n_1} + \frac{S_2{}^2}{n_2}\right)}}$$

Where $\overline{X}$ denotes the mean of the performance criteria being analyzed and S designates its standard deviation. As an example consider the mean flow time of experiment 1, group 1 (product A). See data in table 5.1. The student-test model to compare policy 1 (treatment with the proposed rule, TWK-FB) and policy 2 (treatment with the benchmark rule, IR-TWK) is given by

$$t^1{}_f = \frac{186.39-188.71}{\sqrt{\frac{3.39^2}{10}+\frac{2.63^2}{10}}} = -1.7103$$

The number of degrees of freedom is given by

$$f = \frac{\left(\frac{S_{1_2}}{n_1}+\frac{S_{2_2}}{n_2}\right)^2}{\frac{\left(\frac{S_1^2}{n_1}\right)^2}{(n_1-1)}+\frac{\left(\frac{S_2^2}{n_2}\right)^2}{(n_2-1)}} -2$$

Considering the above example

$$f = \frac{9*(11.49+6.91)}{11.49^2+6.91^2} -2 = 14.954$$

From a t-distribution table it is found that on the basis of 5% level of significance $t_c = t(14.954,0.05) = 1.754$. Thus, as $|t^1{}_f| = 1.7103$ is lower than $t^1{}_f$ whether it is rounded up (1.761) or down (1.753), H_0 is accepted. Consequently, there is no significant difference between the rules when tested on product A in terms of mean flow time at the significance level of 5%.

The statistical results for the remaining performance criteria of all groups are given in appendix 7.

5.2. FIRST EXPERIMENT - FLAT STRUCTURES

Table 5.1 summarizes the results of the first experiment, with the mean flow time depicted as a scatter diagram in figure 5.1.

Mean Flow Time: No significant effect was observed on group 1 (product A). Apart from this, the benchmark rule (IR-TWK) presented significantly better results than the proposed rule (TWK-FB), particularly as the load ratio increases. The exception took place on group 2 (product B), where the proposed rule outperformed the benchmark rule. It can be seen in figure 5.1 that as the load ratio increases the percentage flow time difference between the rules also increases in favour of the benchmark rule. An interesting result took place between products B and L, represented by groups 2 and 4 respectively. Both groups had the same load ratio. The proposed rule outperformed the benchmark rule for group 2 in terms of flow time (2.68% lower), whereas for set 4 the proposed rule presented 1.63% higher flow time than the benchmark one. A possible explanation may be due to the fact that product B presents a flatter BOM than product L. Such a fact is repeated when comparing groups 5 (product M) and 6 (products A and B). Despite having the same load ratio, the proposed rule gave better results for set 6, but not for set 5. As a flat structure, product M just has one vertical level; however, it has more serial operations than the average of A and B combined.

Table 5.1: First Experiment - Flat Structures

SET	RULES	MEAN FLOW TIME	STANDARD DEVIATION - FLOW TIME	WAITING TIME	STANDARD DEVIATION - WAITING TIME	MEAN ABSOLUTE LATENESS	MEAN TARDINESS	% OF TARDY JOBS
A	TWK-FB	186.39	3.39	17.69	2.20	28.68	8.52	25.5
	IR-TWK	188.71	2.63	44.85	2.35	28.17	9.43	32.3
B	TWK-FB	252.25	4.29	45.15	3.67	37.56	22.46	45.4
	IR-TWK	259.20	5.68	88.68	4.27	37.65	25.98	59.1
G	TWK-FB	455.50	9.38	161.00	5.72	165.01	164.96	99.5
	IR-TWK	431.50	8.92	205.99	8.51	141.01	140.95	99.4
L	TWK-FB	244.86	3.84	16.42	1.59	34.62	8.42	23.2
	IR-TWK	240.94	3.70	24.01	1.20	31.89	5.09	24.1
M	TWK-FB	376.44	6.00	109.25	3.00	96.07	94.86	93.1
	IR-TWK	354.57	6.52	136.73	6.10	75.43	73.61	90.9
AB	TWK-FB	273.82	4.59	63.71	3.29	55.39	26.15	41.25
	IR-TWK	274.92	4.05	100.42	4.14	60.28	29.14	41.05
GLM	TWK-FB	582.36	7.02	254.92	22.31	245.36	240.17	87.9
	IR-TWK	522.61	5.43	208.49	51.43	216.76	195.99	70.27
ABGLM	TWK-FB	624.18	8.38	299.57	6.47	190.37	141.11	58.8
	IR-TWK	563.82	6.79	313.81	8.64	196.89	114.19	47.78

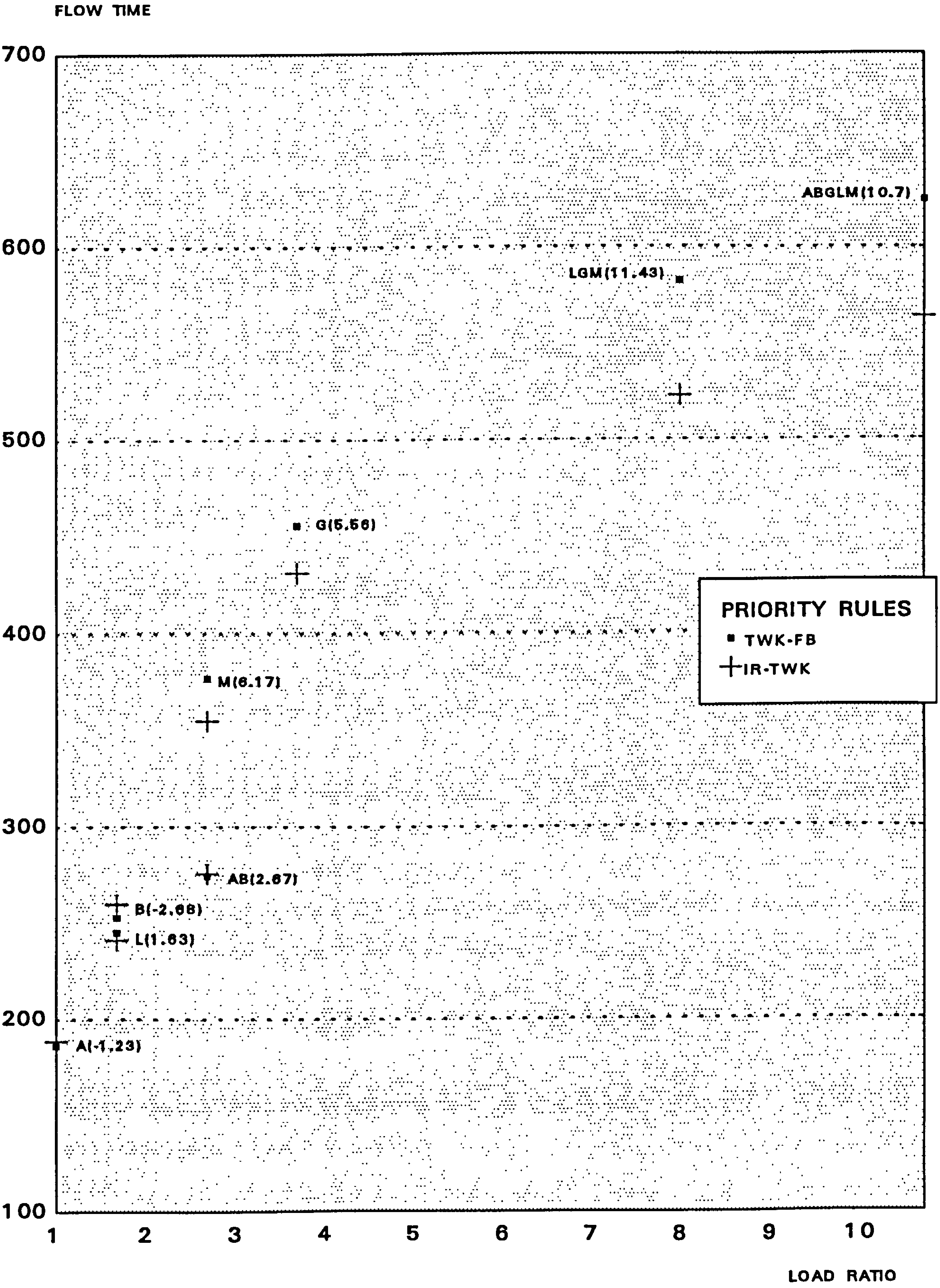


Figure 5.1: Mean Flow Time - Flat Structure

Mean Waiting Time: The proposed rule outperformed the benchmark rule for nearly all flat groups. The only exception took place on group 7 (products G, L and M). Such a result seems to confirm the previous assumption that, by scheduling the critical path in the forward manner and then the non-critical ones in the backward manner, queues are reduced. Figure 5.2 illustrates the percentage differences between rules with strong advantage for the proposed rule.

Mean Absolute Lateness: The proposed rule outperformed the benchmark rule for groups 6 (products A and B) and 8 (products A, B, G, L and M). No significant effect was observed in groups 1 (product A) and 2 (product B). The benchmark rule outperformed the proposed rules in the remaining flat groups.

Mean Tardiness: No significant result was observed in group 1 (product A). However, the proposed rule significantly outperformed the benchmark rule in groups 2 (product B) and 6 (products A and B). The remaining groups favoured the benchmark rule.

Mean Percentage of Tardy Jobs: The proposed rule outperformed the benchmark rule in two groups, group 1 (product A) and 2 (product B). No significant result was noted in groups 3 (product G), 4 (product L) and 6 (products A and B). In the remaining 3 groups, the benchmark rule significantly outperformed the proposed rule.

5.3. SECOND EXPERIMENT - TALL STRUCTURES

The summary results of the second experiment are given in table 5.2. Figures 5.3 and 5.4 describe mean flow time and mean waiting time respectively, in graphical manner.

Mean Flow Time. The proposed rule performed poorly, being outperformed throughout the experiment. The only exception occurred in

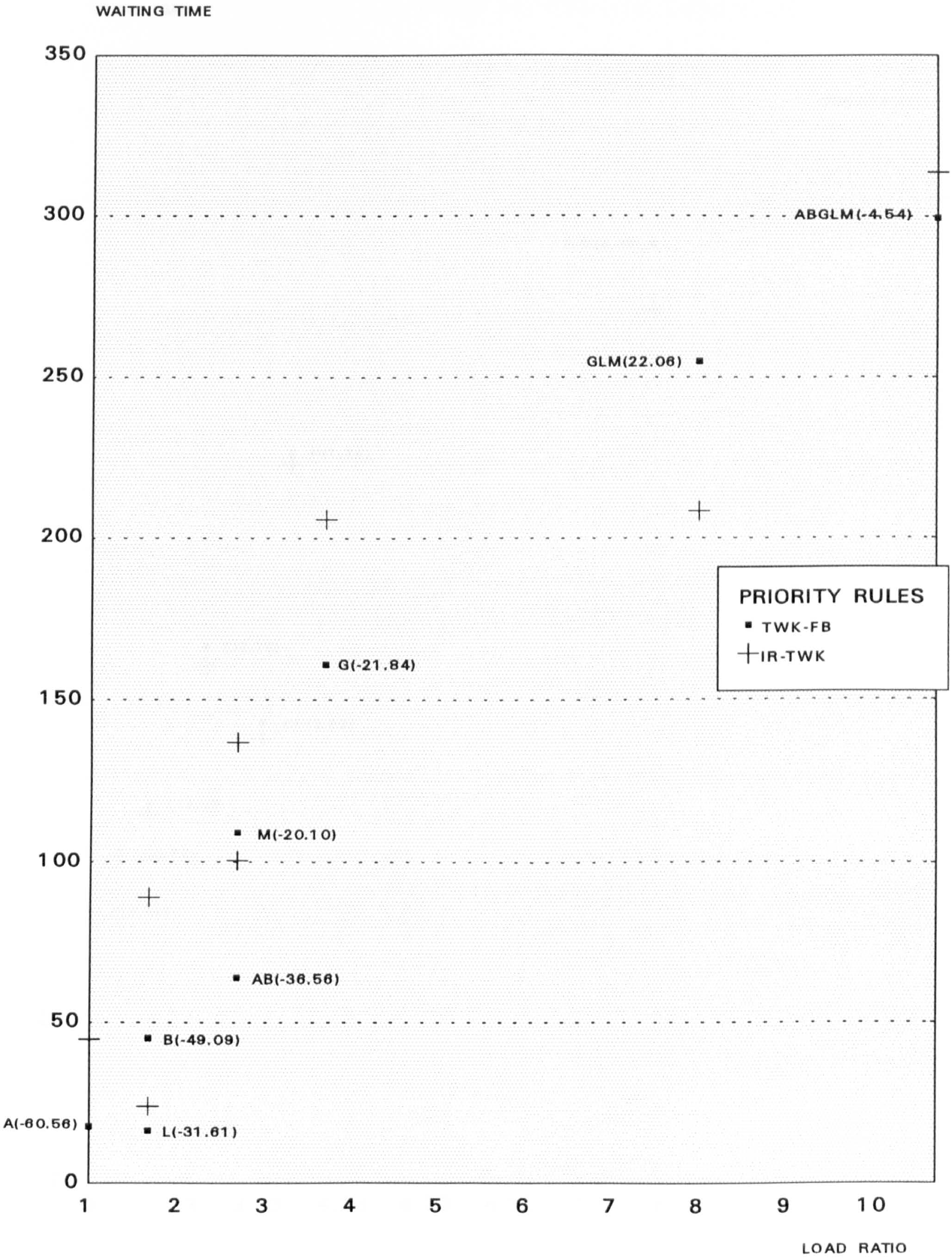


Figure 5.2: Mean Waiting Time - Flat Structure

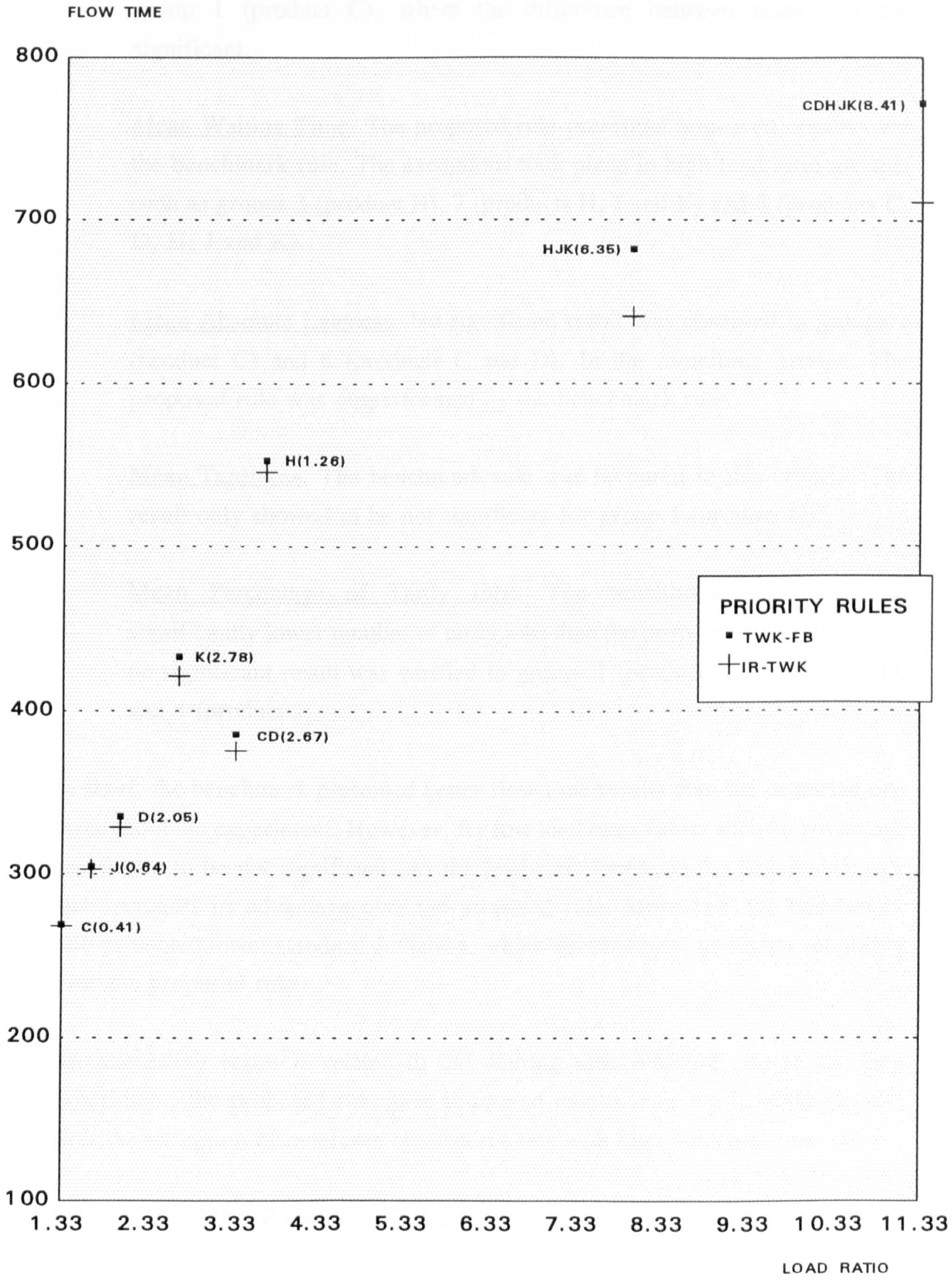


Figure 5.3: Mean Flow Time - Tall Structure

group 1 (product C), where the difference between rules was not significant.

Mean Waiting Time. The proposed rule presented improved results over the benchmark rule. The exceptions took place in high load ratio groups, such as groups 3 (product H), 7 (products H, J and K) and 8 (products C, D, H, J and K).

Mean Absolute Lateness. No significant result was observed in groups 1 (product C) and 6 (products C and D). In the remaining groups, The proposed rule was outperformed by the benchmark rule.

Mean Tardiness. The benchmark rule was favoured in this criteria. The result only showed to be not significant for group 1 (product C).

Mean Percentage of Tardy Jobs. The benchmark rule presented significantly lower number of tardy jobs than the proposed rule. However, no significant result was verified in groups 1 (product C), 2 (product D) and 4 (product J).

In short, the benchmark presented better flow time results than the proposed one throughout the experiment. However, for low load ratio values such an advantage has shown to be not significant. As the load ratio increases the benchmark rule also increases its advantage over the proposed rule. Moreover, the benchmark rule presented lower standard deviation, which demonstrates its higher reliability over the proposed rule.

An interesting result is related to the waiting time measure. As in the first experiment, the proposed rule gave improved results over the benchmark one, with the exception of measures obtained in sets with high values of load ratio.

5.4. ANALYSIS BETWEEN FLAT AND TALL STRUCTURES

Table 5.2: Second Experiment - Tall Structures

SET	RULES	MEAN FLOW TIME	STANDARD DEVIATION - FLOW TIME	WAITING TIME	STANDARD DEVIATION - WAITING TIME	MEAN ABSOLUTE LATENESS	MEAN TARDINESS	% OF TARDY JOBS
C	TWK-FB	268.70	3.13	13.91	1.44	42.10	3.11	10.6
	IR-TWK	267.61	2.70	32.45	4.91	41.54	2.28	10.6
D	TWK-FB	335.38	5.50	53.98	3.10	43.61	28.08	50.3
	IR-TWK	328.64	5.62	69.83	3.95	37.94	21.87	52.8
H	TWK-FB	553.27	6.22	144.77	5.14	65.76	56.38	70.5
	IR-TWK	546.36	4.36	140.06	3.88	61.69	50.89	67.9
J	TWK-FB	304.92	5.89	43.64	2.91	41.12	11.59	25.9
	IR-TWK	302.98	4.44	67.55	3.54	35.58	7.85	28.9
K	TWK-FB	432.75	7.32	96.82	5.92	49.67	28.98	48.6
	IR-TWK	421.05	7.00	107.64	5.28	43.70	19.99	43.7
CD	TWK-FB	385.70	3.17	89.36	3.93	77.30	35.32	42.45
	IR-TWK	375.66	4.54	108.63	3.64	77.39	30.35	38.95
HJK	TWK-FB	681.47	9.69	300.62	8.96	175.63	167.06	82.17
	IR-TWK	640.78	6.98	236.44	3.98	166.47	141.93	66.67
CDHJ K	TWK-FB	771.73	10.38	364.53	7.73	155.05	72.63	46.86
	IR-TWK	711.85	8.60	275.99	8.15	182.05	56.19	35.24

The similarity between patterns of Flat and Tall structures may be noted by comparing the diagrams in figures 5.1 and 5.3. Such a fact is due to the design of the product structures, which deliberately created similar structures in both BOM types. This section compare the performance of two set of twin structures, where the only dissimilarity is related to their BOM types.

Product L is denoted by a Flat structure with 10 operations and has product J as its counterpart in the Tall structure. From figures 5.1 and 5.3, it may be noted that the flow time difference on the tall structure (0.64) is not as pronounced as in the flat structure (1.63). The standard deviation for the tall structure was shown to be higher than for the flat structure. For all the other measures, the pattern between these two structures have kept their similarity.

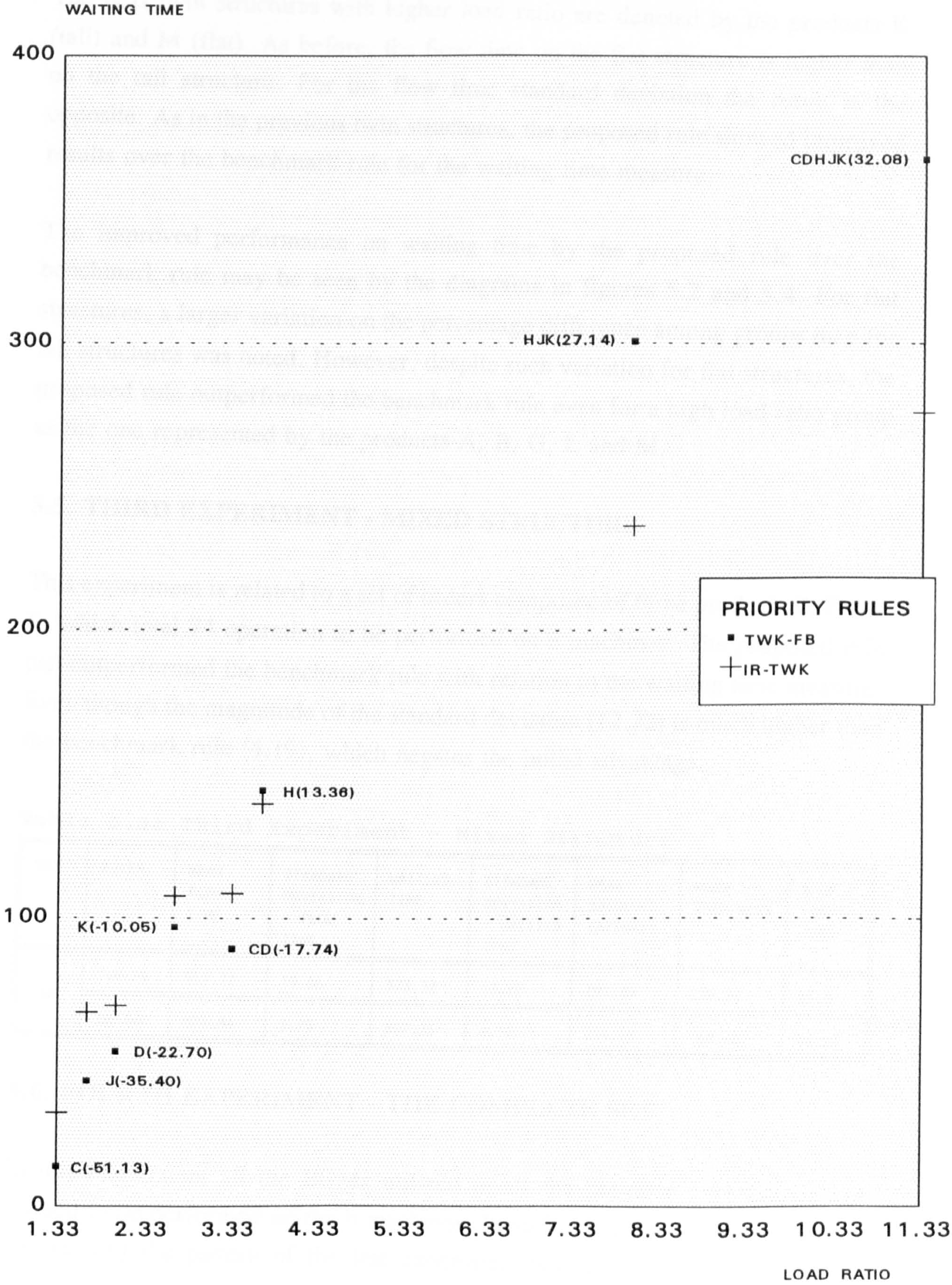


Figure 5.4: Mean Waiting Time - Tall Structure

The next twin structures with higher load ratio are denoted by the products K (tall) and M (flat). As before, the flow time on the flat structure is higher than on the tall structure. For the flow time standard deviation the result is the opposite. As in the previous twin structures, the proposed rule showed improved results over the benchmark rule for the waiting time measure.

The improved performance on waiting time by the proposed rule over the benchmark rule may be seen by the diagrams in figures 5.2 and 5.4. For flat structures, a larger variation on the percentage difference among groups than for tall structures was noted. However, despite such variation for flat structures, the proposed rule outperformed the benchmark rule even for a high load ratio group as the one represented by the products A, B, G, L and M.

5.5. THIRD EXPERIMENT - MIXED STRUCTURE

This experiment is related to a set of orders composed of three products E, F and P, which total 74 operation to be performed on 6 machines. The proposed rule just outperformed the benchmark rule with relation to the waiting time measure. Even though the magnitude of the standard deviation (12.22) is much higher than the benchmark rule (4.19), which negates the initial advantage.

Table 5.3: Third Experiment - Mixed Structure

SET	RULES	MEAN FLOW TIME	STANDARD DEVIATION - FLOW TIME	WAITING TIME	STANDARD DEVIATION - WAITING TIME	MEAN ABSOLUTE LATENESS	MEAN TARDINESS	% OF TARDY JOBS
EFP	TWK-FB	957.37	19.86	373.32	12.22	291.89	275.37	83.6
	IR-TWK	821.84	5.70	397.26	4.19	214.05	168.68	68.87

5.6. FOURTH EXPERIMENT - THE COMPLETE SET

In this experiment all the BOMs utilized in all the previous experiments are queued to be performed on the 6 machines. In such a heavily loaded shop floor ($\alpha=34.33$) the pattern of the last experiment was kept. Now however, the

Table 5.4 Fourth Experiment - All Structures

SET	RULES	MEAN FLOW TIME	STANDARD DEVIATION - FLOW TIME	WAITING TIME	STANDARD DEVIATION - WAITING TIME	MEAN ABSOLUTE LATENESS	MEAN TARDINESS	% OF TARDY JOBS
ALL SETS	TWK-FB	1654.81	77.18	832.19	32.42	646.77	579.07	70.00
	IR-TWK	1426.14	53.67	737.35	29.59	633.63	484.86	60.00

benchmark rule outperformed the proposed rule for all the performance measures, including the waiting time measure.

As it will be formally described in the next chapter, overlapping operations is adopted as an alternative procedure for the scheduling system. In order to prove its value, the same set of experiments was tested on the proposed sequencing rule working with overlapping operations. As was expected the results were quite impressive. The combined rule, the proposed rule plus overlapping procedure, outperformed the benchmark rule for all the performance measures. The simulation results are presented in appendix 8.

5.7. COMPUTATIONAL PERFORMANCE

Usually a sequencing rule is characterized by the time spent in scheduling a number of jobs in a specific number of resources. In the present case, other aspects must also be considered such as (i) the complexity of the BOMs involved in the scheduling process, (ii) the finite capacity procedure, which demands an intensive process of updating capacity information in terms of timing and size of each related resource. Another reason refers to differences on job shop and assembly shop approaches, since a single assembly shop order might encompass a huge number of potential jobs if such an assembly structure was 'exploded' into its components. Considering however, that the current approach is not an optimizing procedure, but a non-backtracking approach, time is not a matter of major concern. In order to provide some information on the speed of the proposed rule table 5.5 describes the computing time spent in scheduling some of the tested groups. Such a test was carried out in a 386SX microcomputer of

16 mhz without maths co-processor.

Table 5.5: Computational Performance

GROUPS	COMPONENTS	ASSEMBLIES	OPERATIONS	COMPUTER TIME (minutes:seconds)
GLM	18	3	48	00:34
HJK	23	10	48	00:36
ABGLM	28	5	64	00:55
CDHJK	35	15	68	01:29
EFP	34	11	73	03:02
ABCDE FGHJK LMNP	97	31	205	23:57

5.8. CONCLUSION

The simulation results suggest that the proposed rule seems to perform better than the benchmark rule for low load ratios and flat structures. Such a conclusion was expected considering the way the proposed rule works. As the vertical levels or number of serial operations increase, the number of rescheduling tasks to allocate operations into time slots also increases. This fact associated with the size of the load ratio caused an increase in the flow time variance. Nevertheless, the proposed rule has given encouraging results in terms of the waiting time performance measure. Such a fact is mainly due to two causes. First, the replacement technique which attempts to schedule operations into time slots that satisfy two conditions: nearness to the immediate operation and time slot size. The second reason is related to the adopted forward and backward approach. By scheduling the non-critical branches in the backward manner after the critical operations have been scheduled in the forward manner, waiting time has been kept under much tighter control than given by the benchmark rule.

Due to the often extremely complex assembly shop structures it could not be said that the obtained results are conclusive. Nonetheless, they provide insights into the applicability of adopting a critical path type sequencing rule for assembly shop. Initially, the proposed rule may be considered as an alternative for scheduling complex assembly structures in environments with low load ratio. Moreover, the adoption of the replacement technique must be re-analyzed in order to investigate the situations where a time slot should be considered. This is because when an operation is rescheduled the cascade effect throughout structure or structures may lead to undesirable results. Another reason lies in the eventuality of changing current schedules already approved. Chapter 6 will consider means of coping with the boundaries of replacement technique.

CHAPTER 6

Finite Capacity Scheduling

This chapter presents the characteristics adopted by the proposed scheduling system such as : The scheduling system encompasses job and assembly shops; the sequencing rule utilized in this study is the TWK-FB rule described in chapter 3; capacity is viewed through each individual resource in terms of their availability intervals; if an availability interval does not suit a given capacity requirement then either rescheduling or batch splitting are considered[10]; overlapping operations may be used either as an ordinary scheduling procedure or as an expediting method.

A realistic scheduling approach has to consider a given production scenario. In other words, the sequencing process is done by considering the current load and capacity on the shop floor, i.e., when a set of orders is to be scheduled, and there are already orders in process. These arriving orders have to adapt themselves to the capacity resultant from the already approved schedules. In the sequencing rules presented in appendix 1, all operations are scheduled in the forward manner. However, if the already mentioned capacity intervals occur, then there must be a way of making use of these availabilities. The current chapter considers the features required by a finite scheduling system, as well as, regarding the adoption of procedures such as *overlapping operations* and *splitting orders*.

It is not an intention of the proposed scheduling system to take into account all the possible situations that may happen on the shop floor, neither to present a

---

[10] In this study, splitting batches is not viewed as an expediting method, but considered for using capacity slots, which otherwise due to their insufficient capacity, would be disregarded.

fully workable system, but demonstrate the feasibility and potentiality of the proposed sequencing rule in a finite capacity environment.

6.1. CHARACTERISTICS OF A FINITE SCHEDULING SYSTEM (FSS)

Many authors have delineated the characteristics of a feasible and flexible scheduling system, e. g., **Graves (1981), Kempf (1989)** and **Rodammer and White (1989)**. In this section some features, particularly related to finite scheduling, are considered.

Initially and above of all, a Finite Scheduling technique is the one that, before scheduling an operation, considers the capacity availability on the resource in which the operation will be performed. Therefore, a FSS checks capacity before scheduling. By having this broad definition as a background the following features are also required.

(i) A FSS must have a constant updating process of the capacity availability database.

The process might be explained as follows: (1) Before scheduling an operation check the availability of its resource according to the requested size (processing time) and requested timing (technological ordering) in the capacity available database; (2) schedule the operation; and (3) update the capacity available database on the regarded resource. Above all, the data utilized must have a high degree of accuracy.

Appendix 3 gives the updating algorithm used throughout this research and figure 1.2 (page 8) illustrates how this updating process relates with the scheduling system.

(ii) The capacity availability database has to express capacity in terms of timing and size. In other words, such a database should be presented as a calendar list of individual resources.

consider the scheduling of the products **R** and **S**, already introduced in section 3.5 (An Example for a Capacity Constrained Environment) and whose product structures are given in figure 3.1 (page 30). Initially, there was plenty of capacity in all the resources related to both orders (table 6.1)[11].

Table 6.1: Initial Capacity Availability

RESOURCE	START TIME	FINISH TIME
a	0.00	∞
b	0.00	∞
c	0.00	∞

Then, the order **R** was selected and scheduled throughout, as depicted in table 6.2.

Table 6.2: Schedule of product **R**

OPERATION	MACHINE	START TIME	FINISH TIME
R10	a	0.00	8.00
R5	a	8.00	13.00
R7	a	13.00	19.00
R3	a	19.00	24.00
R1	a	29.00	34.00
R6	b	0.00	5.00
R8	b	5.00	12.00
R4	b	13.00	18.00
R9	b	19.00	24.00
R2	b	24.00	29.00

When the next order **S** is included in the scheduling process it encounters a capacity availability profile resultant from the scheduling of the previous order

---

[11] The symbol ∞ in table 6.1 denotes an unlimited capacity ahead in time.

(table 6.3).

Table 6.3: Capacity resultant from scheduling the product R

RESOURCE	START TIME	FINISH TIME
a	24.00	29.00
a	34.00	∞
b	12.00	13.00
b	18.00	19.00
b	29.00	∞
c	0.00	∞

The capacity availability profile resultant from the scheduling of S (table 6.4) is given at table 6.5.

As it may be noticed in tables 6.1, 6.3 and 6.5 the availability is related to each resource individually and each one of them, as a timetable, has a starting and completion time. These time limits express the lower and upper availability limits of a resource.

(iii) A FSS should recognize differences on the types of availability intervals.

As an availability interval is generated by the scheduling process itself, it could be said that up to a certain extent, the availability intervals might be seen as a mirror image of the schedule itself. In this fashion, it would be convenient to establish the following differentiation. If a certain set of scheduled jobs have been already approved, then their corresponding capacity intervals are considered to be committed. In that case, to avoid changing the schedule, the rescheduling technique, presented in chapter 3, must not be used. Such a technique is only used if the involved jobs are part of the same scheduling process.

Example: Let us illustrate this concept with the following example.

Table 6.4: Scheduling of the products **R** and **S**

OPERATION	MACHINE	START TIME	FINISH TIME
R10	a	0.00	8.00
R5	a	8.00	13.00
R7	a	13.00	19.00
R3	a	20.00	25.00
S7	a	25.00	33.00
R1	a	33.00	38.00
S4	a	38.00	42.00
S10	a	43.00	46.00
S1	a	57.00	60.00
R6	b	0.00	5.00
R8	b	5.00	12.00
S8	b	12.00	15.00
R4	b	15.00	20.00
R9	b	20.00	25.00
R2	b	25.00	30.00
S5	b	30.00	32.00
S13	b	32.00	34.00
S11	b	37.00	43.00
S9	b	46.00	50.00
S2	b	52.00	57.00
S14	c	27.00	34.00
S12	c	34.00	37.00
S6	c	37.00	44.00
S3	c	44.00	52.00

Suppose a certain number of orders is requested to be scheduled from a given date on a shop floor where it is not allowed to increase the number of shifts nor the number of daily work hours. The scheduling process of this set of orders may have to adjust with scheduling developed previously, but with orders still in process. For such a situation the question is of how to make use of a capacity slot which has an availability less than that which is requested by an arriving operation. Such a situation is not uncommon no matter what type of sequencing rule is utilized by the

Table 6.5: Capacity resultant from scheduling **R** and **S**

MACHINE	START TIME	FINISH TIME
a	19.00	20.00
a	42.00	43.00
a	46.00	57.00
a	60.00	∞
b	34.00	37.00
b	43.00	46.00
b	50.00	52.00
b	57.00	∞
c	0.00	27.00
c	52.00	∞

scheduling system. As an example, consider a situation where an unplanned but profitable order arrives to be scheduled as soon as possible. Suppose further that this order just makes use of 10 hours of a certain machine. By consulting the foreman responsible for the shop floor in which the machine in question is located, it is concluded that there is just 9 hours available. Furthermore, rearranging the current scheduling to adapt it to include the new order is not allowed. The usual procedure would be to 'jump ahead' in order to look for a suitable capacity slot, which would certainly increase the delivery date (figure 3.6, page 47). Section 6.2.2 will consider the possibility of splitting batches in order to make use of availabilities like the one presented above.

6.2. CONSIDERING OVERLAPPING AND SPLITTING PROCEDURES

Capacity restrictions are coped with either by creating capacity or by rationalizing the capacity usage on the shop floor. Capacity is generated by buying equipment, machines and tools, as well as, by contracting workers or increasing the number of daily work hours and shifts. The capacity usage is rationalised by adopting effective sequencing rules, overlapping operations or splitting batches. This section is concerned with the rationalization of the capacity usage, mainly the adoption of overlapping and splitting in a finite scheduling approach for an assembly shop.

6.2.1. OVERLAPPING OPERATIONS

Overlapping has been usually seen as an expediting method to rush orders, which, for one reason or another, will not be able to meet their due-date. The adoption of such a procedure provides a dramatic reduction on lead time, but also increases move and administrative costs. The OPT production system disrupts such a view by yielding scheduling solutions that involve intentional lot splitting and operation overlapping at non-bottleneck work-centres **(Blackstone, 1989)**.

Keeping the same notation used in previous chapters, figure 6.1 depicts the basic configurations related to the overlapping problem. Overlapping happens between two operations and its degree is driven by the operation which has the lowest processing time. According to the approach used in this study the overlapping may take place in both forward and backward passes. Moreover, overlapping can only occur when the batch size contains more than one unit.

Overlapping in the Forward Pass: Figure 6.1.a, b and c illustrate the forward manner, where v and w denote respectively the start and completion times of the scheduled operation which directly-precedes the operation being analyzed (the one which is being regarded for scheduling). In order to differentiate from other reference parameters, W is named *lower testing limit* and V is called *upper testing limit*. Two possibilities may occur. Either the processing time of the operation being analyzed is greater than the processing time of its preceding operation (figure 6.1.b) or the opposite, i.e., the processing time of the operation being analyzed is lower than the processing time of its preceding operation (figure 6.1.c). When an overlapping takes place the reference for the next scheduling shifts from v to v^*. Figure 6.1.b and the equation below define V^*.

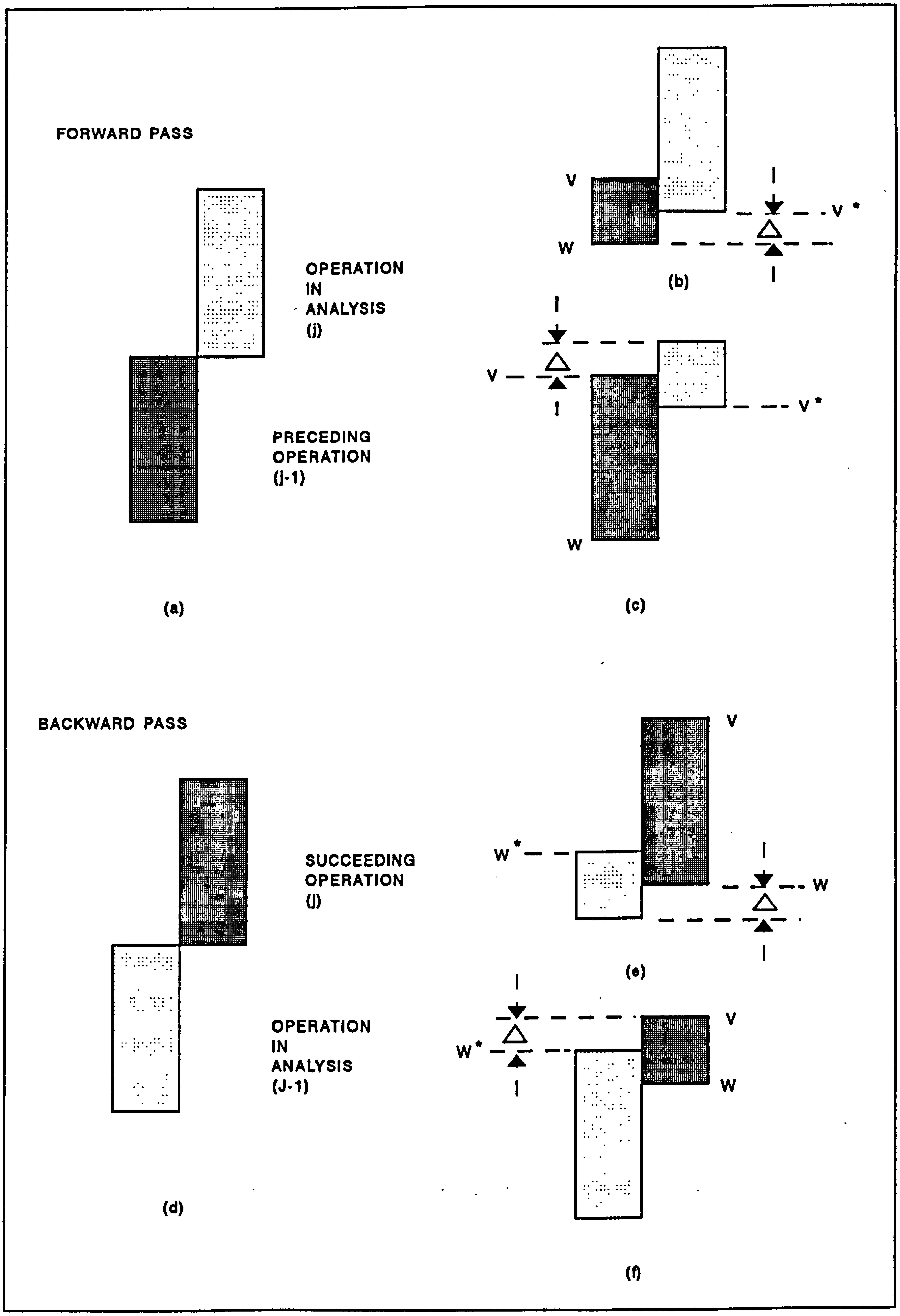


Figure 6.1: Overlapping Configurations

$$V^* = V-(1-\alpha)*Y_{12}*t*u \qquad (6.1)$$

where,

α: overlapping degree, where $0\leq\alpha<1$

$t=\min(B_{1p,j-1,3}, B_{1pj3})$: standard run time

u: number of component units per finished product item of the operation related to t.

The forward manner is characterized by the following relation

j: index of the operation being analyzed

$j-1$: index of the preceding operation

Obviously, the maximum possible overlapping is the one which has a move quantity of one unit only. Let Δ be the production time required by the move quantity q. Accordingly, $\Delta = q*t$. For $q = 1$, then $\Delta = t$. Thus, the overlapping degree is given by $\alpha = \frac{1}{Q}*100$, where Q denotes the batch quantity. Accordingly, the new reference parameter for the maximum overlapping is given by $V^* = V-(Q-1)*t$.

<u>Overlapping in the backward pass</u>: The reasoning used in this case follows the forward case; this time, however, the reference parameter is given by W^*. As before, overlapping operations in the backward manner (figure 6.1.d) is split in two cases: when the processing operation of the operation being analyzed is lower than the processing time of its succeeding operation (figure 6.1.e); and when the processing time of the operation being analyzed is higher than the processing time of its succeeding operation (figure 6.1.f). The new reference parameter W^* for both cases is given by the following equation:

$$W^* = W+(1-\alpha)*Y_{12}*t*u \qquad (6.2)$$

The difference from the forward method is characterized by the relation between operations, where:

j: succeeding operation

$j-1$: operation being analyzed

The effectiveness of overlapping operations has been demonstrated in appendix 8, where the above procedure was combined with the TWK-FB rule and tested against the benchmark rule, IR-TWK. Table 6.6 gives some percentage results when adopting the overlapping procedure on the proposed rule.

Table 6.6: Combining Twk-FB with Overlapping Procedure
PERCENTAGE VALUES

GROUP	MEAN FLOW TIME	MEAN TARDINESS	MEAN ABSOLUTE LATENESS	% OF TARDY JOBS
AB	-16.08	7.92	-52.00	-42.02
CD	-19.69	-60.26	48.33	-53.14
GLM	-4.03	-10.70	-8.98	5.31
HJK	-11.85	-43.02	-27.40	-17.62
EFP	-5.68	-18.58	-7.11	-6.71
ABGLM	-3.22	-6.57	-5.80	1.62
CDHJK	-8.37	-53.50	-0.28	-34.85

6.2.2. SPLITTING BATCHES

Splitting batches is regarded by this study as a capacity usage method in situations where: (i) the capacity available is insufficient; (ii) the scheduling process is not allowed to change previous approved schedules of other orders. **Blackstone (1989)** stresses that batch splitting is a myopic solution to insufficient capacity because it actually leads to a reduction of capacity. This is so because when a batch is split, additional setups are usually required by each generated sub-batch. In this study it is assumed that a new setup is required for each new split batch.

In chapter 3 the concept of the *capacity ratio* (θ) was introduced. As at that stage the only purpose was to compare the performance of the proposed rule against a benchmark rule, many simplifying assumptions were made, among them, the non-existence of setup operations. Now, however, the occurrence of setup[12] is incorporated into the proposed finite scheduling system. Therefore, the processing time (ρ) associated to the order quantity (Y_{12}) is given by the following equation:

$$\rho = B_{1pj4}+Y_{12}*B_{1pj3}*B_{1pj5} \tag{6.3}$$

An element not yet presented is the *number of items required to produce one unit of the final product*, which is denoted by B_{1pj5}.

Another element to be introduced in the array $[B]$ is called *batch size factor* (B_{1pj6}). Such a factor denotes the managerial decision on utilizing a capacity interval that is insufficient to cope with the capacity required. Therefore, instead of looking for another more suitable interval, this factor considers the possibility of using it somehow.

The batch size factor is an integer number that specifies the minimum quantity permissible to be produced for a certain operation. Consider a certain interval that presents a capacity ratio lower than zero. Until now, there were two ways of coping with the interval. First, by ignoring it and searching for another interval. The second alternative was by adopting the rescheduling technique as explained in chapter 3. However, if such an interval is the resultant of an approved scheduling, then to use the rescheduling technique is not advisable since it would change the schedule. Therefore, the question that would probably arise in practice would be 'if the whole requested batch is not possible to be produced then how many items may be produced?' Sometimes such an amount is so much smaller than the original one that the cost involved in setting-up the machine does not justify this alternative. Consequently, it would be convenient to know beforehand the minimum amount permissible for a given operation.

---

[12] The type of set-up used to illustrate the splitting batch approach is sequence-independent, i.e., the technological ordering of operations does not affect the occurrence of setting-up the related task.

Obviously, a batch size factor equal to zero denotes that there is no restrictions on the minimum permissible quantity. In that case, a resource could even be set up to produce just one unit. On the other hand, a high batch size factor would indicate severe restrictions on setting up machines.

This study considers the inclusion of such a factor by defining it as: "The processing time (ρ_{min}) required by the minimum permissible amount (q) must be B_{ipj6} times the time spent to setup its operation (B_{ipj4})." This statement may be expressed as:

$$\rho_{min} = B_{ipj6} * B_{ipj4} \tag{6.4}$$

The minimum processing time may also be expressed as in equation 6.3.

$$\rho_{min} = q * B_{ipj3} * B_{ipj5} \tag{6.5}$$

By combining equations 6.4 and 6.5, the minimum permissible quantity is given by

$$q = \frac{B_{ipj6} * B_{ipj4}}{B_{ipj3} * B_{ipj5}} \tag{6.6}$$

Moreover, equation 6.6 in 6.4 yields

$$\rho_{min} = B_{ipj4} + (1 + B_{ipj6}) \tag{6.7}$$

This study considers the batch size factor as a given parameter regardless of how it was defined. The definition of such a factor may be due to a mere managerial decision or any more sophisticated decision-making alternatives such as a lot-sizing calculation to establish a minimum economical lot.

> Example: A certain machine has an interval of 10 units of time. The operation being analyzed demands a standard run time of 0.2 units of time per unit produced and a setup time of 5 units of time per batch. Moreover, the batch size factor of this operation is equal to 2. Is it possible to utilize such an availability slot to produce 100 units considering that the rescheduling technique is not allowed? The solution is as follows: the total capacity required is 25 units of time, which implies in a capacity ratio equal to 10/25. Therefore, as $\theta < 1$, the capacity available is insufficient to load all the requested quantity. The possibility of using such an interval by

a split operation is restricted to the batch size factor of 2. According to the previous equation the minimum quantity is equal to:

$$q = \frac{2*5}{0.2*1} = 50\,(units)$$

Therefore, the operation being analyzed is split into two sub-batches in order to make use of such an interval once it is capable of accommodating the setup task and the production of the minimum permissible amount (50 units).

6.3. THE FINITE SCHEDULING SYSTEM

Figure 6.2 depicts the basic modules of the proposed system. This section describes such a system through each of its modules. A new example composed by the products **I** and **T** is utilized to describe the scheduling logic. Their BOMs are illustrated in appendix 4.

6.3.1. SETTING UP THE ENVIRONMENT[13]

The flowchart in figure 6.3 supplies the setting up sub-stages. The setting up stage aims to define the arrays which will be used throughout the scheduling process. There are three major types of input:

(i) the user's input, which contains details on the orders to be scheduled, such as product code, quantity and dates;

(ii) the Bill of Resources (BoR) database in which all information on product structures and routings are contained;

(iii) the Resource Availability (RsC) database, which contains a list of availability for each individual resource within a pre-defined period.

The three types of information given above are organized into arrays (A, C, D and E). Such arrays, except array $[A]$, are intermediate steps, which aims to define

---

[13] Some of the information presented in this section have been already introduced in chapter 3. Even though, they are repeated here for the sake of clarity.

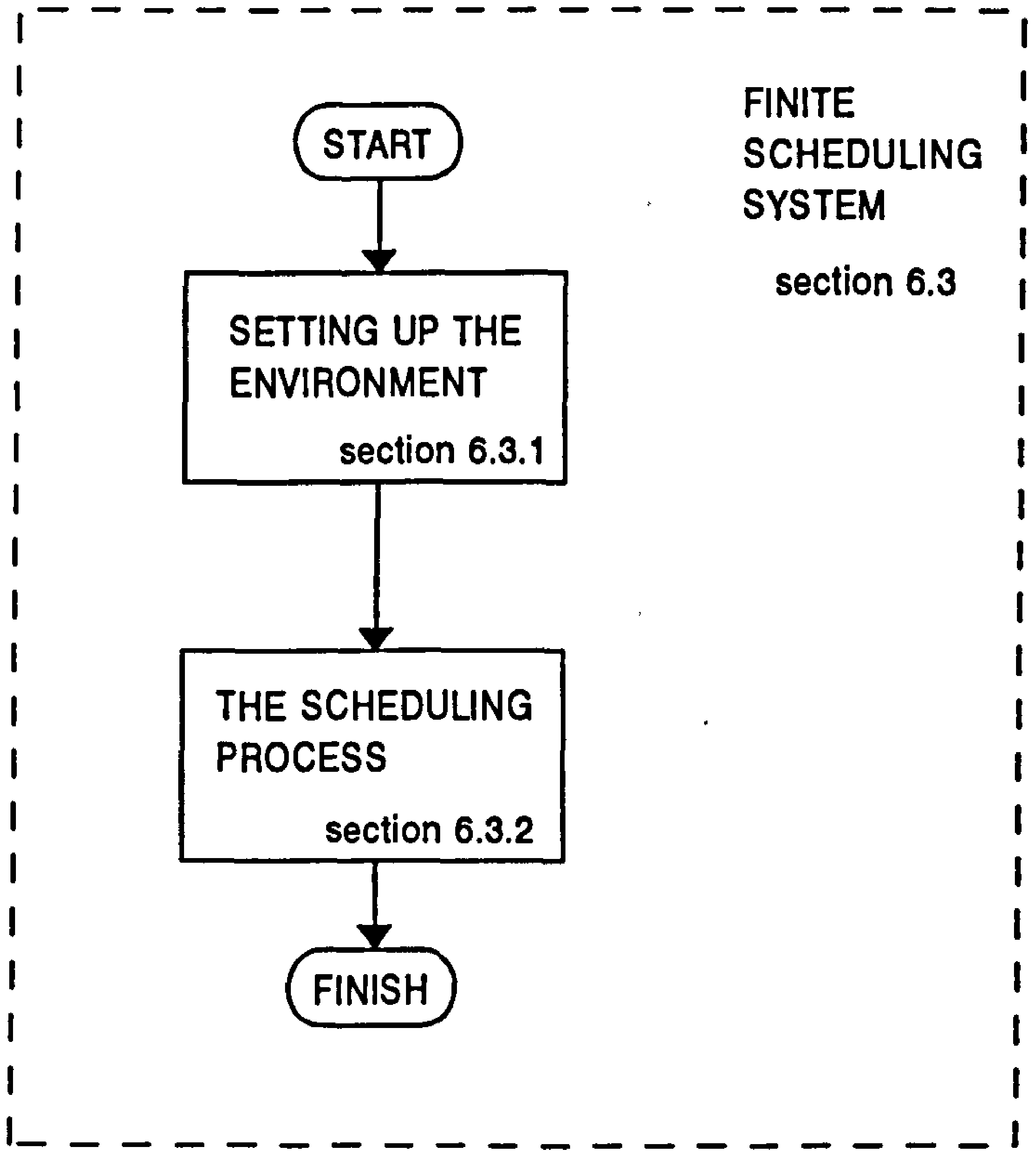


Figure 6.2: System Overview

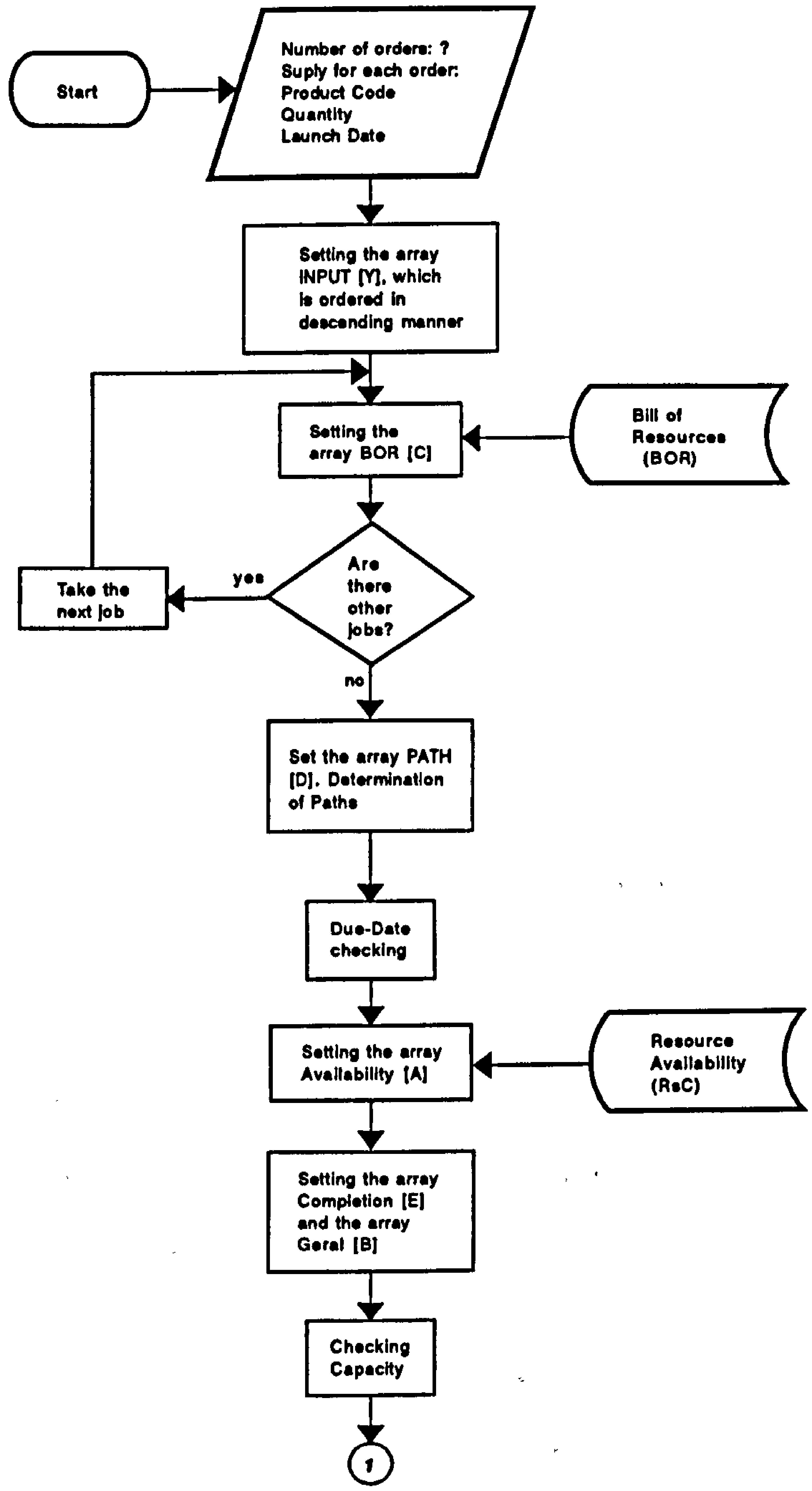


Figure 6.3: Setting Up the Environment

the general array `[B]`. The array `[B]` contains all the required information by the user's input from the BoR database. Such information are properly sorted by the characteristics imposed by the proposed sequencing rule (TWK-FB). Jobs are sorted in an ascending manner since, according to the proposed rule, jobs are prioritized by the least total work content. For each selected job, paths are prioritized by the longest path. In this fashion, it could be said that the array `[B]` reflects the structures of the involved products and their routings, as well as, the characteristics imposed by the proposed sequencing rule.

The user's input is organized in the array $Y[n_i \times 6]$. Thus, it keeps coherence with the network notation adopted in chapter 3. Where

Y_{i1} product code related to order i, where $i \in (1,...,n_i)$

n_i: the maximum number of orders

Y_{i2} quantity of the i^{th} order

Y_{i3} ready time of the i^{th} order

If the parameters given above are inputs, the following ones are dependent variables, since they are calculated during the scheduling process:

Y_{i4} due date[14] of the i^{th} order

Y_{i5} total number of paths related to order i

Y_{i6} total number of operations related to order i

Y_{i7} Total Work Content of the order i

Example: Table 6.7 illustrates the array $[Y]$ with the input to schedule 300 units of the product I and 600 units of the product T, whose structures are depicted in appendix 4.

---

[14] : The due-date is estimated according to the total work content related to the length of the longest path in the BOM of the order i. This method of assigning due-date was defined in chapter 4, section 4.3.

Table 6.7: Input Example

ORDER	PRODUCT	QUANTITY	LAUNCH DATE	DUE DATE	TOTAL No. OF PATHS	TOTAL No. OF OPERATIONS	TOTAL WORK CONTENT
1	I	300	4				
2	T	600	60				

All relevant information are kept in two main databases, i.e., the *Bill of Resources (BoR)* and the *Resource Availability (RsC)*. As the name suggests, the BoR database keeps information on the Bill of Materials of all products manufactured by the company, as well as their routings and operation standard times. The RsC database is actually a calendar of availability interval of each individual resource. Obviously such database undergoes a constant updating process with a high level of accuracy.

Example: Table 6.8 portrays part of the BoR database and table 6.9 supplies part of the RsC database. Each of the columns represent a field of the file structure.

Table 6.8: Bill of Resources (BoR) database

RECORD	PROD_CODE	SUCESS	PREDECESS	MACH_CODE	RUN_TIME	SETUP_TIME	CONTRIB	LOTSIZEFCT
199	Y	Y3	Y6	d	3.00	10.00	1	3
200	I	X	I1	b	0.06	1.0	1	2
201	I	I1	I2	a	0.04	2.0	2	2
202	I	I1	I3	c	0.04	1.0	1	3
203	I	I2	RM					
204	I	I3	I4	c	0.02	1.0	1	2
205	I	I4	RM					
206	I	I3	I5	d	0.04	1.0	1	1
207	I	I5	RM					
208	T	X	T1	b	0.02	1.0	1	1
209	T	T1	T2	a	0.06	1.0	1	1
210	T	T2	T3	b	0.02	1.0	1	0
211	T	T3	RM					
212	V	X	V1	f	0.07	2.0	2	3

X: denotes a dummy operation with the aim to indicate the point from which the calculation starts
RM: a dummy operation, which denotes the raw material utilized by the parent operation
Therefore the assembly structure is a tree with many branches (denoted by RMs) and just one X (denotes the final operation).

It would be unpractical to handle all information from both databases during the

Table 6.9: Resource Capacity Database

RECORD	MAQ	INICIO	FINAL
1	a	4.00	26.00
2	a	32.00	36.00
3	b	25.00	35.00
4	c	4.00	25.00
5	c	30.00	9999.00
6	c	4.00	21.00
7	c	22.00	24.00
8	c	26.00	34.00
9	a	38.00	9999.00
10	b	56.00	9999.00
11	c	36.00	9999.00

scheduling process. Thus, two arrays are created to maintain only the data actually relevant to the scheduling in question. The arrays $A[n_k\ x\ 4]$ and $C[n_c\ x\ 8]$ relate RsC and BoR databases respectively. The former has been mentioned in chapter 3, the latter contains the following information:

C_{c1} product

C_{c2} successor operation

C_{c3} predecessor operation

C_{c4} resource

C_{c5} run time

C_{c5} setup time

C_{c6} contribution: number of units of each component per unit of finished product

C_{c7} batch size factor

c operation index, where $c \in (1,...,n_c)$

The array C contains information on product structure and routing of each involved order. However, for scheduling purposes it is required to designate more clearly their routings. This is done through the array $D[n_i \; x \; n_p \; x \; n_j]$, where n_i is the maximum number of orders, n_p is the total number of paths of all involved orders and n_j is the maximum number of elements (operations) of all paths considered. The array D describes the product explosion into its paths as it was depicted in figure 3.5 (page 45). Concepts such as scheduling through the *longest* path of an order are an integral part of the critical path analysis approach adopted by the proposed sequencing rule (TWK-FB). Therefore, the array calculated above is ordered according to the size of their paths length.

The length of each path, the processing time sum of all operations in a specific path, is placed in the array $E[n_i \; x \; n_p \; x \; 2]$. The two last columns are related to the number of operations and the length of each path respectively.

> Example: Table 6.10 illustrates the arrays [D] and [E] for orders I and T. Where, path p=1 has the longest path length followed by p=2 and then, the shortest path p=3.

Table 6.10: Example for arrays [D] and [E]

[D]

	j	1	2	3
i	p			
1	1	I2	I1	
1	2	I5	I3	I1
1	3	I4	I3	I1
2	1	T3	T2	T1

[E]

i	p	1	2
1	1	2	44
1	2	3	40
1	3	3	38
2	1	3	181

At this stage the first checking takes place. As the completion time of each path is already calculated, by having the launch date of each order as a reference the due-date of each order may be checked. Figure 6.4.a depicts the computer message of the due-date checking stage. Supposing the due-date of a specific order is not able to be met, the user is given the option either of interrupting the scheduling process or permitting continuation of the simulation. In that case, unless the user directs the scheduling to be carried out by overlapping, the order in question will be certainly tardy.

The second checking level verifies if the capacity available is sufficient to satisfy the capacity requirement within the limit established by the given launch date and the calculated due-date. It is interesting to note that this checking considers the size but not the timing. Therefore, this checking may be seen as a preliminary and rough capacity check. Figure 6.4.b provides the computer message of a scheduling run that had not the required capacity. As in the first checking level, the user may continue with the simulation or interrupt it. An alternative should be the continuation of the scheduling process by using the overlapping mode.

The setting up stage culminates in the creation of the multidimensional array $[B]$, which by acting as a node, contains all the information necessary for the scheduling process. All the above arrays are just preliminary stages to model the array $[B]$. In chapter 3, the array $[B]$ was introduced as containing three types of information, i.e., operation code, resource code and standard run time. Now, however, the number of required information are extended from 3 to 6 as follows:

B_{ipj1}: operation

B_{ipj2}: resource

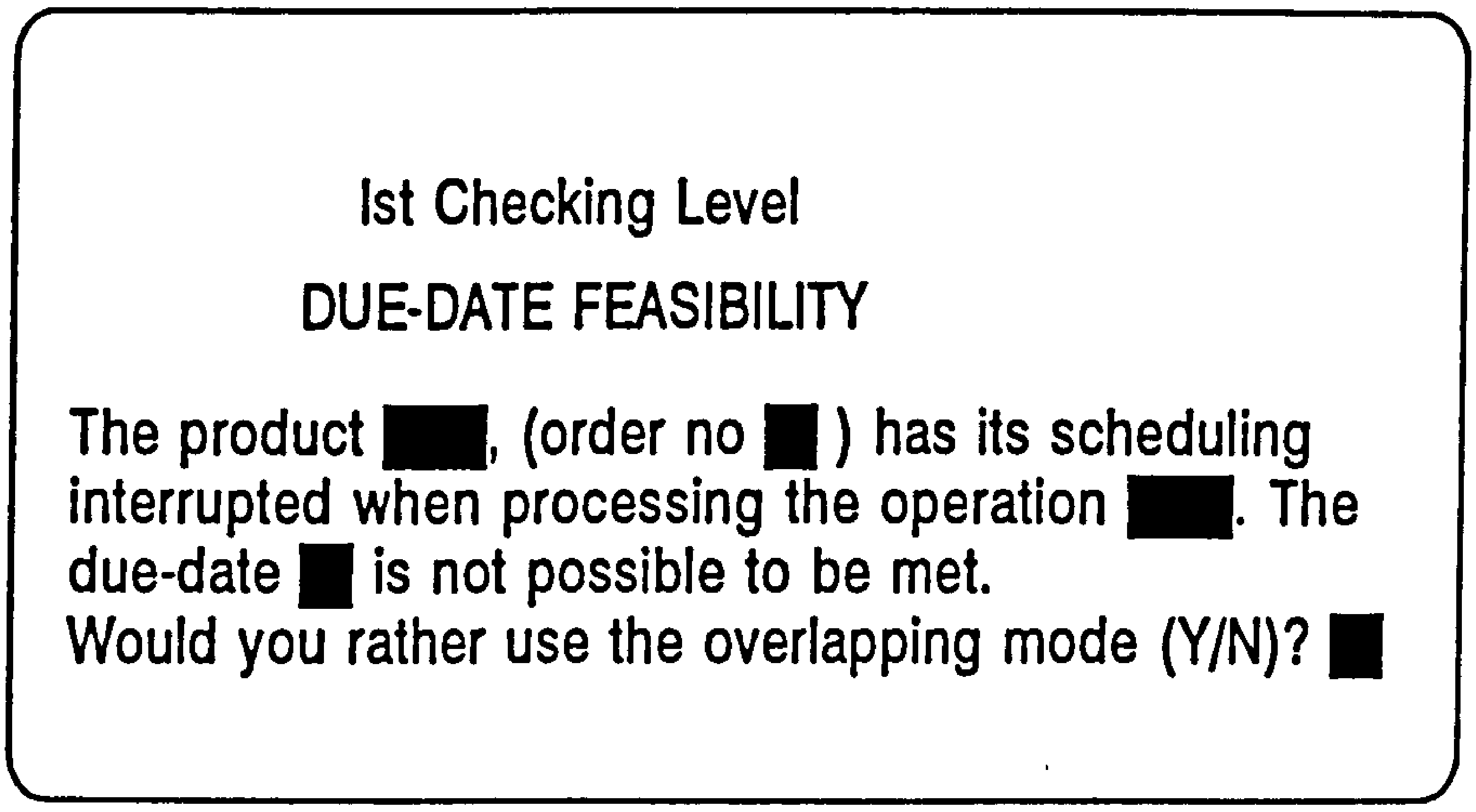
B_{ipj3}: standard run time

B_{ipj4}: standard setup time

B_{ipj5}: number of items required to produce one unit of the final product.

B_{ipj6}: batch size factor

Example: Table 6.11 illustrates the array $[B]$ with the orders **I** and **T**. A relevant particularity of the array $[B]$ is related to its sorting: the first column is sorted in an ascending manner according to the work content (TWK) of each job. Note that jobs are selected according to the least total work content.

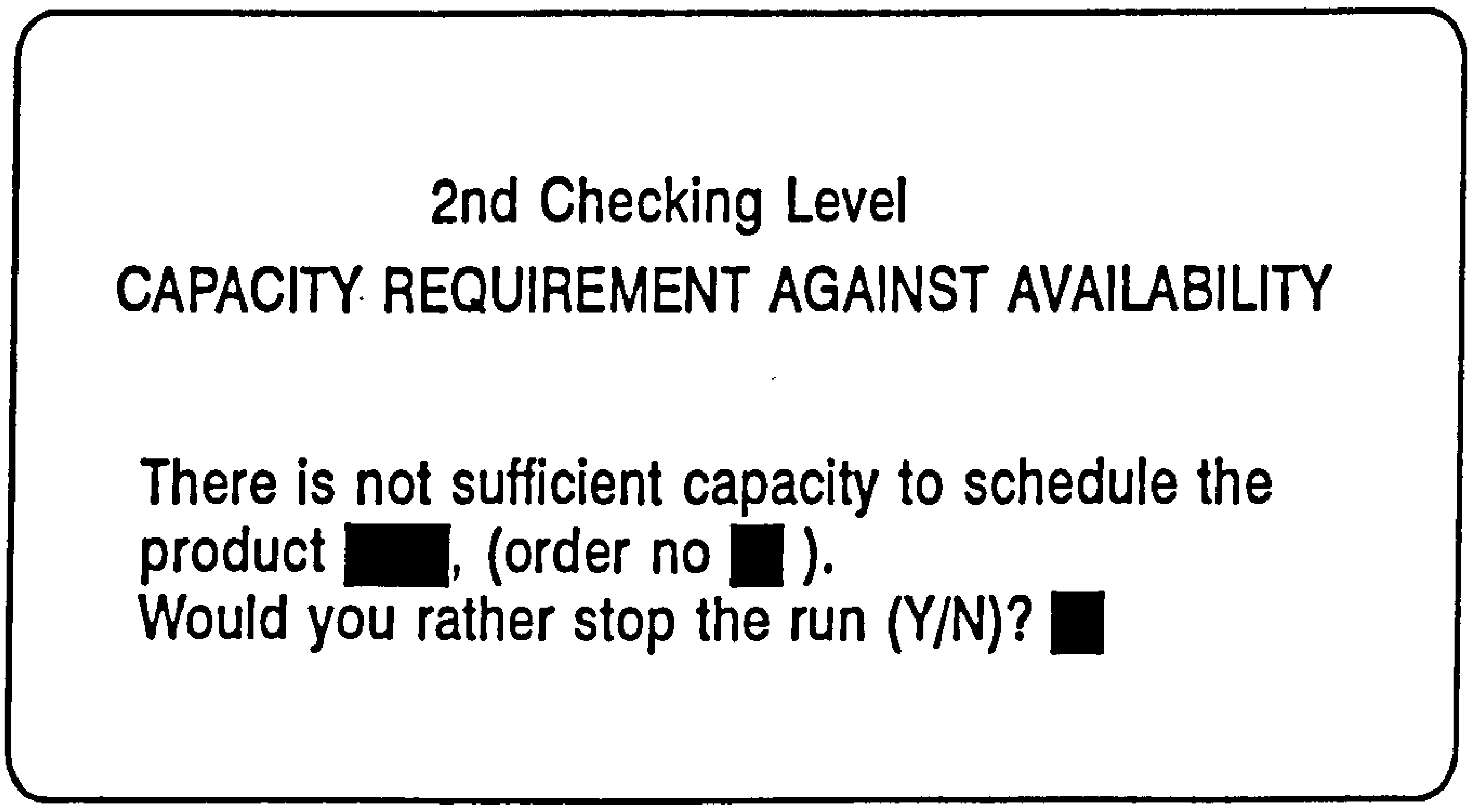


Ist Checking Level

DUE-DATE FEASIBILITY

The product ■, (order no ■) has its scheduling interrupted when processing the operation ■. The due-date ■ is not possible to be met.
Would you rather use the overlapping mode (Y/N)? ■

a) Due-Date Feasibility



2nd Checking Level

CAPACITY REQUIREMENT AGAINST AVAILABILITY

There is not sufficient capacity to schedule the product ■, (order no ■).
Would you rather stop the run (Y/N)? ■

b) Capacity Feasibility

Figure 6.4: Computer Messages

Table 6.11: The General Array [B]

i	p	j	1	2	3	4	5	6
1	1	1	I2	a	0.04	2.00	2	2
1	1	2	I1	b	0.06	1.00	1	2
1	2	1	I5	d	0.03	1.00	1	1
1	2	2	I3	c	0.04	1.00	1	3
1	2	3	I1	b	0.06	1.00	1	2
1	3	1	I4	c	0.02	1.00	1	2
1	3	2	I3	c	0.04	1.00	1	3
1	3	3	I1	b	0.06	1.00	1	2
2	1	1	T3	b	0.06	1.00	1	1
2	1	2	T2	a	0.18	1.00	1	1
2	1	3	T1	b	0.06	1.00	1	0

Furthermore, within the job, paths are sorted according to their lengths in a descending manner. In this fashion, those jobs with longer lengths are selected first.

The last task of the setting up stage is to set the array that will receive the schedule, $F[n_f\ x\ 6]$.

where

F_{f1} Operation related to the element index f

F_{f2} Denotes the type of component related to the operation: run or setup component operation

F_{f3} Resource

F_{f4} Starting time

F_{f5} Completion time

F_{f6} Number of units scheduled by the operation in analysis

n_f Number of schedule elements

The objective is to determine the set of starting and completion times for each operation (schedule element): $\{F_{f4}\ \forall f \in (1,...,n_f)\}$ and $\{F_{f5}\ \forall f \in (1,...,n_f)\}$.

6.3.2. THE SCHEDULING PROCESS

Once the environment is set up the actual scheduling process may start. This section demonstrates the scheduling process through the example of orders I and T given above. The heuristic approach follows the logic depicted in the flowchart in figures 6.5.a to 6.5.c.

Step 1. The Total Work Content (TWK) of each order involved is calculated and the results are placed in column 7 of the array y. Where, TWK is defined as the processing time sum of all operations in a particular job. The array [*Y*] is then sorted in an ascending manner. In this fashion, orders with lower TWK are prioritized. Set job index i to 1 and go to the next step.

example: From both orders in queue, the order related to 300 units of the product I has the least total work content. Then, it is the chosen one.

$$TWK_I = 300*(0.06+0.04*2+0.04+0.02+0.03)+6 = 75$$
$$TWK_T = 600*(0.06+0.18+0.04)+3 = 171$$

Step 2. This step verifies if the order selected to be scheduled is part of the list of orders (array Y) by asking if $i \le n_i$, where n_i is the total number of orders in the present simulation. If so, go to step 3 to finish the scheduling process.

Example: Two orders to be scheduled, then $n_i = 2$

Step 3. Here, a number of variables and parameters is set to their initial values. Set:

path: $p=1$

lower testing limit: $W=0$

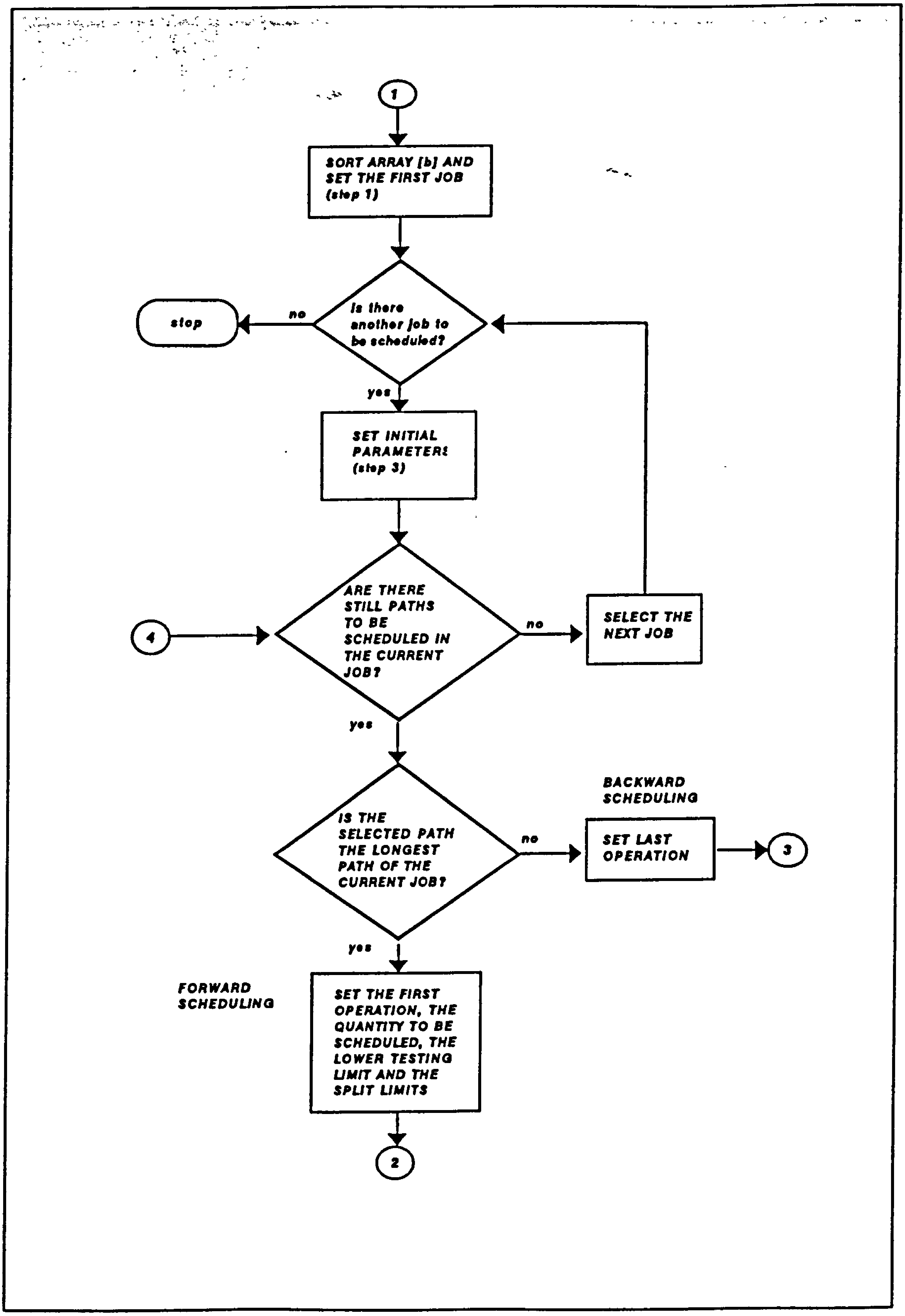


Figure 6.5a: The Proposed Finite Scheduling Approach – Part 1

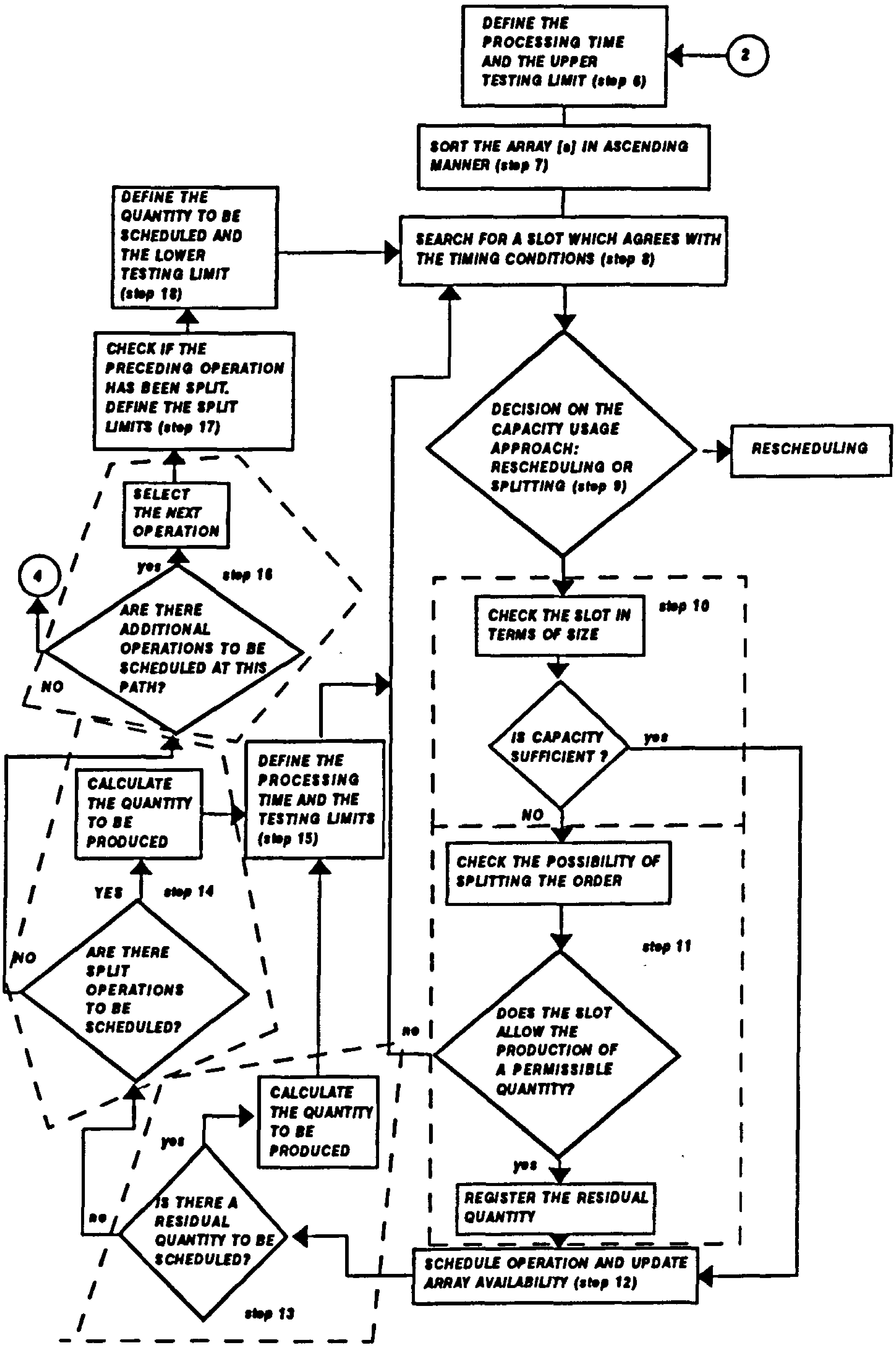


Figure 6.5b: The Proposed Finite Scheduling Approach – Part 2

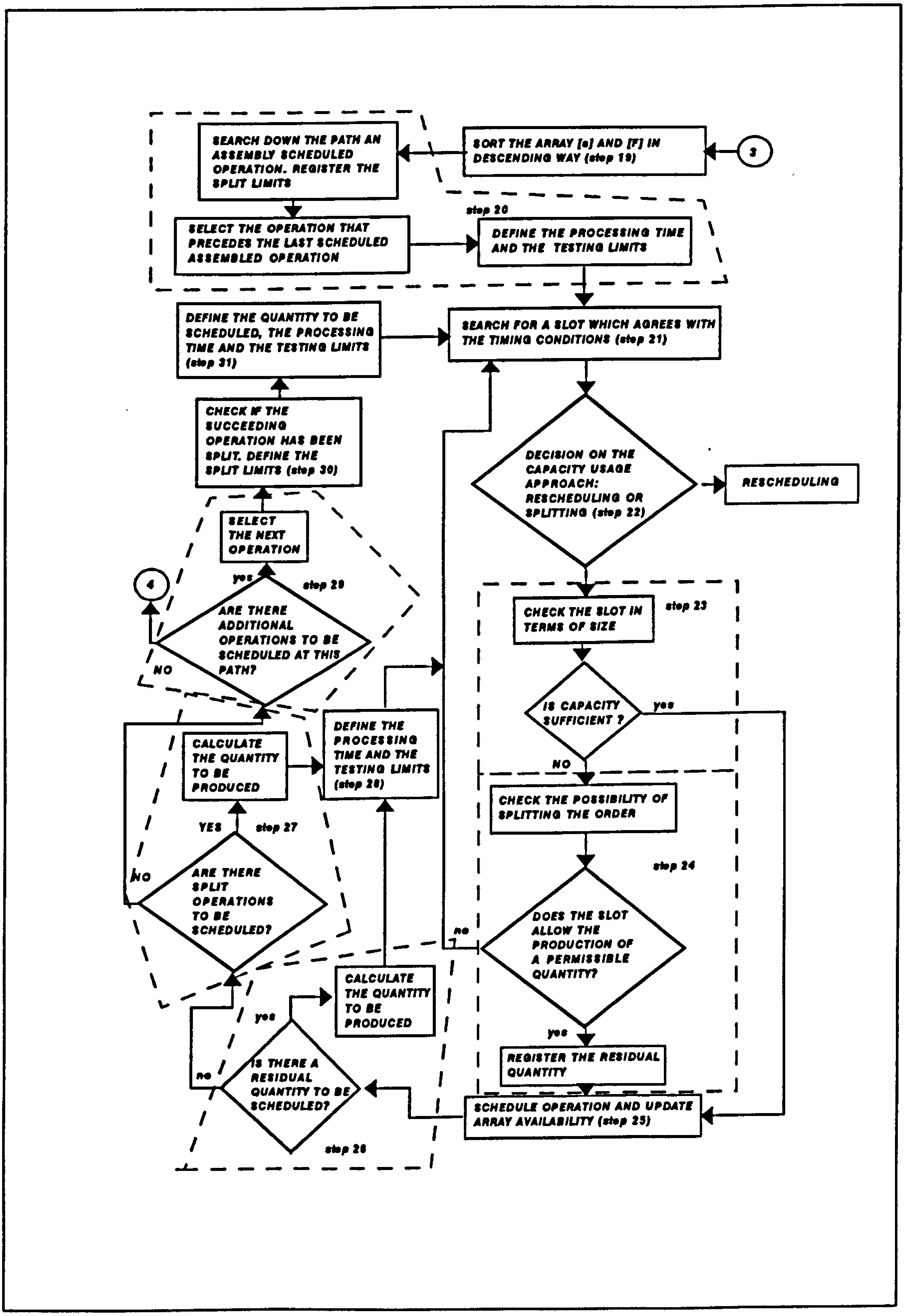


Figure 6.5C: The Proposed Finite Scheduling Approach – Part 3

upper testing limit: $V=0$

scheduling index: $f=1$

Go to step 4.

Step 4. This step checks if there are paths yet to be scheduled in the current job by asking if the selected path belongs to the set of paths to be considered, i.e, if $p \leq Y_{i5}$. If so, as in the example case, go to the next step. Otherwise, select another job by increasing i by one and return to step 2.

Step 5. This step determines if the current path is to be scheduled in the forward or in the backward manner. If $p=1$, which denotes the longest path, then this path is to be scheduled in the forward manner. In that case, set:

the first operation as $j=1$,

the quantity to be scheduled as $Q=Y_{i2}$,

the lower testing limit as $W=Y_{i3}$ and,

the split limits as $\lambda=\omega=1$.

Go to step 6. Otherwise, if the selected path is not the longest one ($p \neq 1$), set the last operation of the current path as $j=E_{ip1}$ and go to step 19.

<u>Example</u>: The lower testing limit is set equal to the launch date, i.e., $W=4$. The quantity to be scheduled is equal to the order quantity, i.e., $Q=300$.

Step 6. Define the processing time ρ related to the operation being analyzed B_{ipj1}. Where $\rho=B_{ipj4}+Q*B_{ipj3}*B_{ipj5}$. This step also defines the upper testing limit V. Where, $V=W+\rho$. Go to step 7.

<u>Example</u>: $\rho=2+300*0.04*2=26$ and $V=4+26=30$.

Step 7. Sort the array availability [A] in the ascending manner. The references for the sorting are the first column (resource code) and the second column (initial date). This procedure guarantees that the intervals will be sequentially selected from the most recent to the farthest slot.

Example: At this stage the array [A] presents its initial configuration, which is sorted in an ascending manner as depicted in table 6.12:

Table 6.12: Array [A] sorted in ascending way

INDEX k	1 RESOURCE	2 START TIME	3 FINISH TIME	4 FLAG
1	a	4.00	26.00	LABEL
2	a	32.00	36.00	LABEL
3	a	38.00	∞	
4	b	25.00	35.00	LABEL
5	b	56.00	∞	
6	c	4.00	25.00	LABEL
7	c	30.00	∞	
8	d	4.00	21.00	LABEL
9	d	22.00	24.00	LABEL
10	d	26.00	34.00	LABEL
11	d	36.00	∞	

Step 8. This step checks the feasibility of an interval in terms of timing according to the following conditions:

$$A_{k1} = B_{ipj2}$$
$$A_{k3} > W$$

Go to the next step.

Example: The first interval which comprises with the timing conditions is denoted by $k=1$.

Step 9. This step analyzes the type of approach to utilize the capacity available, rescheduling or splitting. If A_{k4}=*LABEL* that means the slot was generated for approved scheduled operations. Therefore, it can be used as long as it does not alter the start and completion dates of the operations already scheduled. The opposite situation, i.e., when the slot has no flag, denotes the possibility of rearranging the whole schedule, approved or not. Such a situation is considered just among arriving orders[15].

Step 10. The size of the selected capacity interval is checked by the capacity ratio θ. Where $\theta = \frac{A_{k3}-A_{k2}}{V-W}$. If $\theta<1$, then the capacity is insufficient to accommodate the load. If so, go to the next step, otherwise go to step 12.

Example: The capacity is considered insufficient, once $\theta=\frac{26-4}{30-4}=0.85$.

Step 11. The fact that the capacity slot in question is not large enough to accommodate the required load does not mean that such a slot will be disregarded. This step analyzes the possibility of splitting the order to make use of the slot in question. Calculate the minimum quantity (q) permissible to be produced at this particular operation (B_{ipj1}) by means of the following equation:

$$q = \frac{B_{ipj4} \cdot B_{ipj6}}{B_{ipj3} \cdot B_{ipj5}}$$

Moreover, calculate the maximum quantity ($\bar{q}$) capable of being produced in such a slot.

---

[15] As the *Replacement Technique* was already introduced in chapter 3, for the sake of clarity, this algorithm will just make use of the splitting method by assuming that all capacity slots have received flags.

$$\bar{q} = \frac{A_{k3}-A_{k2}-B_{ipj4}}{B_{ipj3}.B_{ipj5}}$$

If $\bar{q} > q$, then the slot is utilized and the order is split. In that case, register the remaining quantity (ξ), which is not able to be scheduled at the current interval, as $\xi = Q-\bar{q}$. Set $Q=\bar{q}$ and go to step 12. If the previous condition is not satisfied, return to step 8.

Example: $q=\frac{2*2}{0.04*2}=50$ and $\bar{q}=\frac{26-4-2}{0.04*2}=250$. As $\bar{q}>q$, then the operation is split. The remaining quantity is given by $\xi=300-250=50$ and Q is set as 250.

Step 12. This step schedules the operation being analyzed, as well as updates the array [A]. Initially, however, it is necessary to verify if the last operation scheduled at the current resource is similar to the operation about to be scheduled, $B_{ipj1}=F_{f1}$ for $B_{ipj2}=F_{f3}$ and $F_{f5}=W$. If so the setup operation is not considered. Then, go to step 12.2, otherwise go to the next step. A similar occurrence may take place if the current operation does not require a setup ($B_{ipj4}=0$). In that case, the procedure is similar.

Step 12.1. Increase f by one. The scheduling of the setup operation is given by:

$$\begin{aligned}
&F_{f1}=B_{ipj1}\\
&F_{f2}=SET\\
&F_{f3}=B_{ipj2}\\
&F_{f4}=W\\
&F_{f5}=W+B_{ipj4}\\
&F_{f7}=\lambda
\end{aligned}$$

Step 12.2. Increase f by one and schedule the run operation as follows:

$$
\begin{aligned}
&F_{f1}=B_{ipj1}\\
&F_{f2}=RUN\\
&F_{f3}=B_{ipj2}\\
&F_{f4}=V\\
&F_{f5}=V+Q*B_{ipj3}*B_{ipj5}\\
&F_{f6}=Q\\
&F_{f7}=\lambda
\end{aligned}
$$

The upper testing limit is updated as $V=F_{f4}$. Appendix 3 supplies details on the updating module.

Increase λ by one and go to the next step.

Example: Tables 6.13 and 6.14 presents the array schedule [F] and updated array availability [A] respectively. After the updating both arrays are re-sorted in an ascending manner.

Table 6.13: Schedule of operation I2 sorted in an ascending manner

INDEX f	1 OPERATION	2 TYPE	3 RESOURCE	4 START TIME	5 FINISH TIME	6 QUANTITY	7 BATCH No.
1	I2	SETUP	a	4.00	6.00		1
2	I2	RUN	a	6.00	26.00	250	1

Step 13. If there is a remaining quantity to be scheduled ($\xi>0$), define $Q=\xi$ and go to step 15, otherwise go to the next step.

Example: The quantity to be considered is equal to the remaining quantity as $Q=\xi=50$.

Step 14. This step checks if there are still split operations to be scheduled, which is true if $\lambda\leq\omega$. If so, define $Q=F_{f6}$ and go to step 15. Otherwise, go to

Table 6.14: Updating the array [A] after scheduling I2

INDEX (k)	1 RESOURCE	2 START TIME	3 FINISH TIME
1	a	32.00	36.00
2	a	38.00	∞
3	b	25.00	35.00
4	b	56.00	∞
5	c	4.00	25.00
6	c	30.00	∞
7	d	4.00	21.00
8	d	22.00	24.00
9	d	26.00	34.00
10	d	36.00	∞

step 16

Step 15. The processing time and testing limits are defined at this stage as follows:

$$\rho = B_{ipj4}+Q*B_{ipj3}*B_{ipj5}$$
$$W=V$$
$$V=W+\rho$$

Return to step 8.

At this stage, since the procedure return to previous steps, the example continues till it reaches the step 15

Example:

$$\rho = 2+50*0.04*2=6$$
$$W = 26$$
$$V = 26+6=32$$

To schedule the second batch of 50 units at the operation I2, the interval (32.00, 36.00), denoted by k=1, is considered. However, such a interval is neither sufficient to accommodate the requested load nor a permissible amount, as it can be seen from the following calculations:

$\theta = \frac{36-32}{32-26} = 0.67 < 1$, and the permissible quantity is given by

$q = \frac{2*2}{0.04*2} = 50$ units. However, the interval just allows the production

of $\bar{q} = \frac{(36-32)-2}{0.04*2} = 25$ units. Therefore, the next availability interval of the resource a is selected. Such an availability interval is sufficient to accommodate the residual quantity of the operation I2. Consequently the residual amount of I2 is scheduled at this interval in array **F** and the availability array [A] is updated. Both arrays are sorted as specified previously and the scheduling process goes to step 15 since there is no more remaining quantity to be scheduled.

Step 16. This step regards the possibility of having additional operations to be scheduled in the current path which is done by asking if $j>E_{ip1}$. If so, return to step 4. Otherwise select the next operation by increasing j by one and go to the next step.

example: As the current path has two elements (I2 and I1). Thus, there is still one element to be scheduled, i.e., $B_{ipj1}=I1$.

Step 17. Here, the preceding operation is scanned for occurrence of splitting. Such an inspection is done by searching at the array of scheduled operations [F] for all operations that are similar to the preceding operation $B_{ipj-1,1}$. Obviously, if the preceding operation has been split, then it would have more than one operation similar to itself. At the beginning of the searching process the indexes λ and Ω are set to f. The upper split limit Ω is increased by one whenever a successful match occur, i.e., $F_{f1}=B_{ipj-1,1}$. In the end, the indexes λ and Ω denote the limits of the splitting operation by identifying at the array [F] the first and the last split operations that are similar to the preceding operation. Let $f=1$, if $F_{f1}=B_{ipj-1,1}$ set $\lambda=\Omega=f$, increase f by one and repeat the previous comparison. For each new successful match increase Ω by one.

Example: Table 6.15 presents a sorted array [F]. As the preceding operation is I2, the lower split limit λ is related to $f=1$ and the upper split limit is denoted by $f=4$.

Table 6.15: Sorted array [F] - splitting batches

f INDEX	1 OPERATION	2 TYPE	3 RESOURCE	4 START TIME	5 FINISH TIME	6 QUANTITY	7 BATCH No.
1	I2	SET	a	4.00	6.00		1
2	I2	RUN	a	6.00	26.00	250	1
3	I2	SET	a	38.00	40.00		2
4	I2	RUN	a	40.00	44.00	50	2

Step 18. From the last step it may be concluded that if $\Omega-\lambda\leq 1$ than no splitting has taken place. Besides, the quantity to be scheduled is placed in the 6th column of scheduled run operations only. Therefore, the determination of the quantity to be scheduled is done according to the following procedure: If $F_{\lambda 2}=RUN$ set $Q=F_{\lambda 6}$. If $F_{f3}\neq B_{ipj2}$ set $W=F_{\lambda 5}-B_{ipj3}$, otherwise let $W=V$. The calculation of W guarantees superposition between the setup of the operation being analyzed and the preceding operation. In this fashion it is not necessary to wait the whole preceding operation to finish in order to start setting up the next operation. If $F_{\lambda 2}\neq RUN$, increase λ by one and repeat the previous comparison. This procedure is followed while $\lambda\leq\Omega$. Return to step 6

As the scheduling process returns to previous steps, the example continues until the current path is concluded.

Example: From table 6.15 $Q=250$. As the resource utilized to perform the operation being analyzed (b) is different from the one (a) used by the preceding operation, let W=26−1=25. The upper testing limit is given by V=25+250∗0.06∗1=40. The slot (25.00, 35.00) is considered. However, such a slot denotes insufficient capacity since $\theta<1$. On the other hand, such a slot is utilized once it is able to produce 150 units, which is higher than the minimum permissible of 33 units. Accordingly, the remaining quantity (ξ) is determined as 100 units. Therefore, 150 units are scheduled at operation I1 and the slots related to the resource in analysis are updated. The remaining quantity is scheduled by regarding the next slot (56.00, 9999.00), which provides sufficient availability ($\theta>1$). The Gantt chart on the left in figure 6.6 supplies details of the scheduling, as well as of the capacity configuration till this point. As no remaining quantity is requested the next split operation is considered by increasing λ by one. For such a split operation ($F_{\lambda 1}$=*I2* and $F_{\lambda 7}$=2) Q=50 and W=44−1=43. The upper testing limit is defined as V=50. The selected slot (63.00, 9999.00) provides sufficient capacity ($\theta>1$). The scheduling of the second batch of I1 finds out that the last operation, which was scheduled on the resource in analysis, is similar to the operation being analyzed. In that case, it is not necessary to alter the setup of the resource in analysis by repeating the same setup. The complete forward scheduling of the current path is depicted in the next Gantt Chart in figure 6.6.

Step 19. This step denotes the beginning of the backward scheduling process. The first task is to sort both arrays [F] and [A] in the descending manner. In this fashion, it is guaranteed that all searches start from operations and availability slots, which are scheduled most forward in time. Go to the next step.

Step 20. The first element in the present path that has not been scheduled is selected in order to be scheduled next. Its immediately succeeding path element, which was already scheduled, is considered since it denotes an assembly

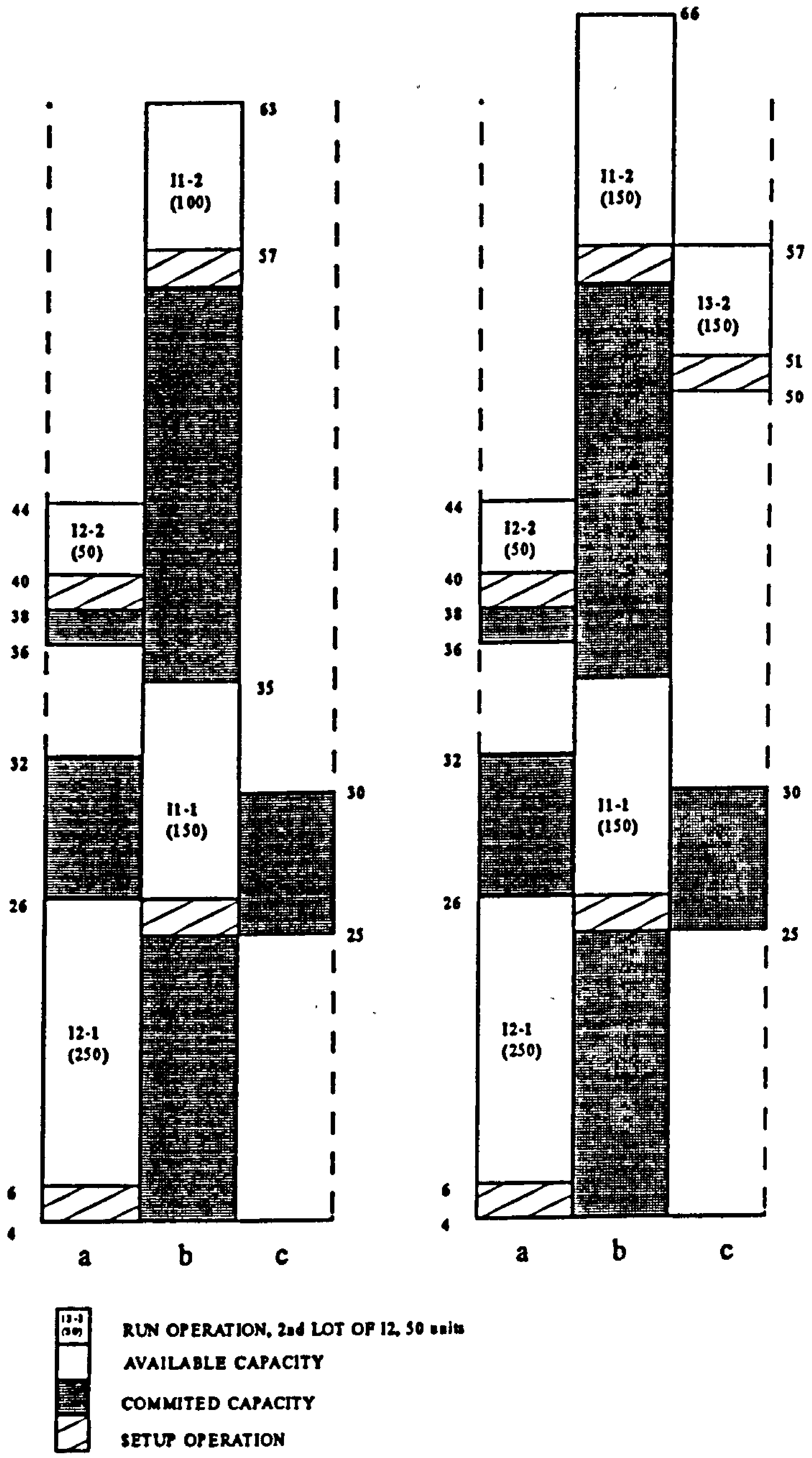


Figure 6.6: Intermediate Results – Gantt Chart

operation[16]. During the checking process details on the operations scheduled are registered. Consequently, registering details on the assembly operation as well. Moreover, the occurrence of splitting on the scheduled operations in analysis is also considered by registering λ and Ω. In addition, it is convenient to consider that the searching for the assembly operation only happens between paths, but not within a path. The searching for the assembly operation is also corroborated by the fact that a non critical path always contains elements which were already scheduled. The only exception is the critical path, which is the first to be scheduled. Once an assembly operation is identified all the following path elements, which have their calculations carried out within the path, are driven by the technological sequence of operations. Set f=1, if $F_{f1}=B_{ipjl}$ set $\lambda=\Omega=f$. Otherwise increase f by one and resume this step. If $F_{f2}=SET$, set $W=F_{f4}$ and increase f by one. If $F_{f2}=RUN$ set $V=F_{f5}$ and $Q=F_{f6}$. After all scheduled operations are checked, decrease j by one and resume the present step. Repeat this procedure while there exist matching operations ($F_{f1}=B_{ipjl}$). As soon as an operation being analyzed (B_{ipjl}) does not find a similar operation already scheduled, then the operation to be scheduled is determined.

> Example: The path selected is I5>>I3>>I1. As I1 has been already scheduled in another path it denotes an assembly operation. The operation being analyzed is the directly-preceding operation not yet scheduled, I3. As the assembly operation has been split into two sub-batches the splitting limits are defined as $\lambda=1$ and $\Omega=4$. The processing time $\rho=1+150*0.04*1=7$ and testing limits are $V=57$ and $W=57-7=50$. Table 6.16 illustrates the schedule accomplished so far.

---

[16] As all non-critical paths contains at least one element that is also part of the critical path, it is assured that there exists at least one operation, which has been already scheduled, that is similar to the operation being analyzed. Accordingly, such an operation is an assembly operation since it is the succeeding operation of two operations at least: the one from the critical path and the other from the non-critical path being analyzed.

Table 6.16 Intermediate scheduling after finishing the forward pass - Sorted in a descending manner

f INDEX	1 OPERATION	2 TYPE	3 RESOURCE	4 START TIME	5 FINISH TIME	6 QUANTITY	7 BATCH No
1	I1	RUN	b	57.00	66.00	150	2
2	I1	SET	b	56.00	57.00		2
3	I1	RUN	b	26.00	35.00	150	1
4	I1	SET	b	25.00	26.00		1
5	I2	RUN	a	40.00	44.00	50	2
7	I2	SET	a	38.00	40.00		2
8	I2	RUN	a	6.00	26.00	250	1
9	I2	SET	a	4.00	6.00		1

Until this stage the example given followed the algorithm steps. From this point, the algorithm continues without the aid of the example since several steps would be required to be repeated. Consequently, the example would not be as didactic and practical as expected. However, table 6.17 and 6.18 depict the final scheduling and capacity profile for both orders.

Step 21. This step checks the feasibility of an slot in terms of timing according to the following conditions:

$$A_{k1} = B_{ipj2}$$
$$A_{k2} < V$$

Step 22. This step decides if the way of making use of the selected slot is through rescheduling or splitting (see footnote 16 in page 104).

Step 23. The size of the selected capacity slot is checked by the capacity ratio θ. Where $\theta = \frac{A_{k3}-A_{k2}}{V-W}$. If $\theta<1$, then the capacity is insufficient to accommodate the load. If so, go to the next step, otherwise go to step 25.

Table 6.17: Final Schedule - Orders I and T

RESOURCE	OPERATION	INITIAL	FINAL
a	I2-SET-1	4.00	6.00
	I2-RUN-1	6.00	26.00
	I2-SET-2	38.00	40.00
	I2-RUN-2	40.00	44.00
	T2-SET	78.00	79.00
	T2-RUN	79.00	115.00
b	I1-SET-1	25.00	26.00
	I1-RUN-1	26.00	35.00
	I1-SET-2	56.00	57.00
	I1-RUN-2	57.00	66.00
	T3-SET	66.00	67.00
	T3-RUN	67.00	79.00
	T1-SET	114.00	115.00
	T1-RUN	115.00	127.00
c	I4-SET-1	14.00	15.00
	I4-RUN-1	15.00	18.00
	I3-SET-1	18.00	19.00
	I3-RUN-1	19.00	25.00
	14-SET-2	46.00	47.00
	I4-RUN-2	47.00	50.00
	I3-SET-2	50.00	51.00
	I3-RUN-2	51.00	57.00
d	I5-SET-1	12.00	13.00
	I5-RUN-1	13.00	19.00
	I5-SET-2	44.00	45.00
	I5-RUN-2	45.00	51.00

Step 24. The fact that the capacity slot in question is not large enough to accommodate the requested load does not mean that such slot will be disregarded. This step analysis the possibility of splitting the order to make use of the slot in question. Calculate the minimum quantity (q) permissible to be produced at this particular operation B_{ipjl} by means of the following equation:

$$q = \frac{B_{ipj4} * B_{ipj6}}{B_{ipj3} * B_{ipj5}}$$

Table 6.18: Final Capacity Configuration

RESOURCE	INITIAL	FINAL
a	32.00	36.00
a	44.00	78.00
a	115.00	∞
b	79.00	114.00
b	127.00	∞
c	4.00	14.00
c	30.00	46.00
c	57.00	∞
d	4.00	12.00
d	19.00	21.00
d	22.00	24.00
d	26.00	34.00
d	36.00	44.00
d	51.00	∞

Moreover, calculate the maximum quantity ($\bar{q}$) capable of being produced in such a slot.

$$\bar{q} = \frac{A_{k3} - A_{k2} - B_{ipj4}}{B_{ipj3} * B_{ipj5}}$$

If $\bar{q} > q$, then the slot is utilized and the order is split. In that case, register the remaining quantity (ξ), which is not able to be scheduled at the current slot as $\xi = Q - \bar{q}$, set $Q = \bar{q}$ and go to step 25. If the previous condition is not satisfied, return to step 21.

Step 25. This step schedules the operation being analyzed, as well as updates the array **a**.

Step 25.1. Increase f by one and schedule the run operation as follows:

$$F_{f1}=B_{ipj1}$$
$$F_{f2}=RUN$$
$$F_{f3}=B_{ipj2}$$
$$F_{f5}=V$$
$$F_{f4}=V-Q*B_{ipj3}*B_{ipj5}$$
$$F_{f6}=Q$$
$$F_{f7}=\lambda$$

It is still necessary to verify if the last operation scheduled at the current resource is similar to the operation about to be scheduled, $B_{ipj1}=F_{f1}$ for $B_{ipj2}=F_{f3}$ and $F_{f5}=W$. If so the setup operation is not considered. Then, go to step 25.3, otherwise go to the next step. A similar occurrence may take place if the current operation does not require a setup ($B_{ipj4}=0$). In that case, the procedure is similar.

Step 25.2. Increase f by one. The scheduling of the setup operation is given by:

$$F_{f1}=B_{ipj1}$$
$$F_{f2}=SET$$
$$F_{f3}=B_{ipj2}$$
$$F_{f5}=F_{f-1,4}$$
$$F_{f4}=F_{f-1,4}-B_{ipj4}$$
$$F_{f7}=\lambda$$

The upper testing limit is updated as $V=F_{f4}$. Go to the next step.

Step 25.3. Appendix 3 supplies details on the updating module. Increase λ by one and go to the next step.

Step 26. If there is a remaining quantity to be scheduled ($\xi>0$), define $Q=\xi$ and go to step 28, otherwise go to the next step.

Step 27. This step checks if there are still split operations to be scheduled, which is true if $\lambda\leq\omega$. If so, define $Q=F_{f6}$ and go to step 28. Otherwise, go to step 29.

Step 28. The processing time and testing limits are defined at this stage as follows:

$$\rho = B_{ipj4}+Q*B_{ipj3}*B_{ipj5}$$
$$V=W$$
$$W=V-\rho$$

Return to step 21.

Step 29. This step regards the possibility of having additional operations to be scheduled in the current path, which is done by asking if $j>1$. If so, return to step 4. Otherwise select the next operation by decreasing j by one and go to the next step.

Step 30. Here, the succeeding operation is scanned for checking if it has been split. Such inspection is done by searching in the array of scheduled operations (F) for all operations that are similar to the succeeding operation $B_{ip,j-1,1}$. Obviously, if the succeeding operation has been split it would have more than one operation similar to itself. At the beginning of the searching process the indexes λ and Ω are set to f. The upper splitting limit Ω is increased by one for each successful match. In the end, the indexes λ and Ω denote the limits of the splitting operation by identifying at the array [F] the first and the last split operations similar to the succeeding operation. Let $f=1$, if $F_{f1}=B_{ip,j-1,1}$ set $\lambda=\Omega=f$, increase f by one and repeat the previous comparison. For each new successful matching increase Ω by one.

Step 31. From the last step it may be concluded that if $\Omega-\lambda\leq 1$ than no splitting has taken place. Besides, the quantity to be scheduled is placed in the 6th column of scheduled run operations only. Therefore, the determination of the quantity to be scheduled is done according to the following procedure: If $F_{\lambda 2}=RUN$ set $Q=F_{\lambda 6}$. If $F_{j3}\neq B_{ipj2}$ set $V=F_{\lambda 4}$, otherwise let $V=W$. The lower testing limit is calculated as $W=V-\rho$, where $\rho=B_{ipj4}+Q*B_{ipj3}*B_{ipj5}$. The calculation of V guarantees superposition between the setup of the operation being analyzed and the preceding operation. In this fashion it is not necessary to wait the whole preceding operation to finish in order to start setting up the next operation. If $F_{\lambda 2}\neq RUN$, increase λ by one and repeat the prior comparison. This procedure is followed while $\lambda\leq\Omega$. Return to step 21.

6.4. ADDITIONAL CONSIDERATIONS

The scheduling process is concluded with the elaboration of reports on the final scheduling, as well as on the resulting capacity configuration. The present approach favours to present the scheduling as a Gantt chart whenever it is feasible and useful. As **Blackstone (1989)** and **Rodammer and White (1989)** point out, despite the widespread use of Gantt charts over the years, due to its user-friendly interface and advances on computer technology, Gantt charts should be an integrant part of any manufacturing management system. At this stage, the proposed scheduling process should ask the user, or have this information from the initial input, if the BoR and RsC databases are to be updated with scheduling and list of availabilities. If the user decides not to alter the related databases, then the scheduling process is to be understood as a simulation study.

This proposal has been presented as a static approach. However, alterations could be made on it in order to adapt it to work as a dynamic scheduling tool.

A final, but paramount consideration is related to the applicability of a Finite Scheduling approach such as the one presented above. As was described at the beginning of this chapter, a FSS demands an intensive updating process. The proposed approach provides an internal updating procedure throughout the scheduling process. However, in practical terms, its is also required to have mechanisms to provide correspondence between the schedule and the actual manufacturing process, for which the plan has been developed. In other words, it is fundamental for the sake of the schedule feasibility that the FSS be actually as real time scheduling. The automatic on-line computerized monitoring capability required by the FSS would, therefore, keep track of eventual discrepancies between what has been planned and what has occurred. As a consequence, measurements could be taken in order to maintain the schedules validity during the production period. Nonetheless, in order to avoid hyper-sensitivity between plan and production, it is necessary to predefine the variation range allowable for the parameters being monitored. When a data is found to be out of the specified range, correction measures may be automatically undertaken to update the related databases and then, to reschedule the current short term production plan (Naji, 1991). Consequently, it could be said that the feasibility of FSS systems is directly related to advances in manufacturing technology. Initially through computer systems able to rapidly handle massive amount of data. Later, by linking the scheduling system to on-line manufacturing devices, such as data collection units and machine readable codes. Judging by recent advances of manufacturing and computer technology[17] many examples of automated manufacturing work-shops are already present with an inexorable trend towards fully integrated manufacturing systems.

---

[17] As an example, refer to the proceedings published in the series *ADVANCES IN PRODUCTION MANAGEMENT SYSTEMS* by Elsevier Science Publisher B. V., Amsterdam.

CHAPTER 7

Conclusion

The objective of this thesis was to investigate the feasibility of adopting a Critical Path Analysis type sequencing rule for an assembly job shop. In order to accomplish such an investigation an assembly shop sequencing rule was designed, as described in chapter 3. Initially, it was verified that applying the Critical Path Analysis method on its own would have limited validity, since situations such as operations competing for limited resources are not necessarily solved by the method. Moreover, the backward approach associated with the capacity constraint environment could provoke anomalies, such as scheduling an operation in the backward manner could cause it to start before the given order launch date. In order to supply feasible schedules, it was necessary to incorporate some additional technique. Therefore, the replacement technique was developed in order to provide feasible schedules and also to make use of time slots of capacity, which otherwise, would not be considered. The initial results were quite encouraging as can be seen from the examples given in chapter 3. However, it was necessary to test the validity of such a rule, named TWK-FB, against another proven and effective assembly priority rule. In chapter 2 a rule named IR-TWK was selected since it outperformed a number of known assembly rules in a number of criteria. To test the effectiveness of the proposed rule a static simulation model was designed in chapter 4 and the results obtained were analyzed in chapter 5. With exception of the waiting time measure, the obtained results did not outperform the benchmark rule, unless the capacity environment presented low levels of load ratio. Nevertheless, the assumption made at the beginning of the chapter 3, prior to the designing of the proposed rule, was achieved. The proposed rule showed improved results in terms of waiting time, specially for low load ratios, which is particularly advantageous in term of work-in-progress reduction. As the scheduling system in chapter 6 would incorporate overlapping operations as a procedure, it was decided to re-simulate the more complex groups from the experiments of chapter 4, by combining the proposed rule with the overlapping procedure. It was demonstrated in table 6.6 (page 85) and appendix 8 that dramatic reductions were achieved, which permitted the proposed rule to outperform the benchmark rule in nearly all the tested BOMs in respect of the utilized performance measures.

One of the merits of the present investigation is the fact that, by scheduling in the backward manner, issues such as capacity restraints had to be considered. Therefore, chapter 6 illustrated the performance of the Critical Path based rule on a production environment, which kept similarity with a real shop floor, in terms of capacity limitations. Despite many simplifications, the system considered aspects such as setup time and capacity time slots. Some guidelines on a feasible finite scheduling system were considered, as well as splitting orders as a method for utilizing capacity slots.

As a final contribution of this investigation, it could be said that it refers to the way capacity is approached. With advances in computer and manufacturing technology, finite capacity has been increasingly considered as a rational method to provide feasible and flexible schedules. Consequently, justifying to approach production capacity in terms of individual resources time slots.

The proposed scheduling system had not any intention of being considered as a complete system. However, it may be seen as a guideline for tailor-made systems, especially for small and medium-sized plants that make products to order.

A recommendation for further study could be to extend the investigation in terms of identifying the boundaries of the applicability of Critical Path rules in terms of product structure and shop floor capacity complexity. Future advances on this approach could even provide an unified approach for project and production scheduling, as far as product structure complexity is considered.

Another recommendation would be the development of a Finite Scheduling system with broader characteristics and capabilities in order to satisfy real-life shop floor demands. In this context such a system should consider the recommendation delineated by **Philipoom** ***et al.*** **(1991)** of creating a library of proven and effective assembly shop heuristic rules. The scheduling system could automatically select the most suitable rule to a specific situation from a collection of priority rules based on knowledge of the type of job, shop condition and criteria to be satisfied. In this knowledge base library, the proposed rule would have a well defined niche in terms of its applicability: assembly shops with low load ratios when the work-in-progress criteria is the predominant one. The versatility of such a system would effectively satisfy the multiplicity of real assembly shops.

REFERENCES

ADAM, N. R., BERTRAND, J. W. M. and **SURKIS, J.**, 1987, Priority Assignment Procedures in Multi-Level Assembly Job Shops, *IIE TRANSACTIONS*, **19**, 317-327

AKERS, S. B. and **FRIEDMAN, J.**, 1955, A Non-Numerical Approach to Production Scheduling Problems, *Operations Research*, 3, 429-442

AWANI, A. O., 1983, *Project Management Techniques*, (New York: Petrocelli Books, Inc.)

BAKER, K. R., 1974, *Introduction to Sequencing and Scheduling*, (New York: John Wiley)

BATTERSBY, A., 1970, *Network analysis for planning and scheduling*, (New York: St. Martin's Press)

BEDWORTH, D. D. and **BAILEY, J. E.**, 1987, *Integrated Production Control Systems* (New York: John Wiley)

BLACKBURN Jr, J. H., 1986, Considerations in Choosing Shop Scheduling Algorithms, *Production and Inventory management*, 4th quarter, 107-125.

BLACKSTONE Jr, J. H., PHILLIPS, D. T. and **HOGG, G. L.**, 1982, A State-of-the-art Survey of Dispatching Rules for Manufacturing Job Shop Operations, *International Journal of Production Research*, 20, 27-45

BLACKSTONE Jr, J. H., 1989, *Capacity Management*, (Ohio: South-Western Publishing Co.)

BUXEY, G., 1989, Production scheduling: Practice and Theory, *European Journal of Operational Research*, 39, 17-31

CAMPBELL, H. G., DUDEK, R. A., and **SMITH, M. L.**, 1970, A Heuristic Algorithm for the N Job, M Nachine Sequencing Problem, *Management Science*, **16**, 630-637

CHENG, T. C. E., 1990, Analysis of Material Flow in a Job Shop with Assembly Operations, *International Journal of Production Research*, 28, 1369-1384.

CONNER, J. L., 1972, A Heuristic Method for Solving Jobshop Sequencing Problems, *Production and Inventory Management*, **13**, 54-68.

CONWAY, R. W., **MAXWELL, W. L.** and **MILLER, L. W.**, 1967, *Theory of Scheduling*, (Reading, Mass: Addison-Wesley).

COOPER, P., 1972, The Application of Planning Networks in Workshop Scheduling, in *INTERNET 72 - 3rd International Congress*, Stockholm.

CORKE, D. K. 1977, *Production Control in Engineering*, (London: Edward Arnold).

DAVIS, E. W., 1973, Project Scheduling Under Resource Constraints - Historical Review and Categorization of Procedures, *AIIE TRANSACTIONS*, **5**, 297-313

ELMAGHRABY, S. E., 1967, On the Expected Duration of PERT Type Networks, *Management Science*, **13**, 299-306

ELSAYED, E. A. and **BOUCHER, T. O.**, 1985, *Analysis and Control of Production systems*, (New Jersey: Prentice-Hall Inc.)

FRY, T. D., **OLIFF, M. D.**, **MINOR, E. D.** and **LEONG, G. K.**, 1989, The Effects of Product Structure and Sequencing Rules on Assembly-Job Shop Performance, *International Journal of Production Research*, **27**, 671-686

GALGUT, P. E., 1978, Computer-Aided Production Control, *College of Manufacturing lecture notes*, Cranfield Institute of Technology, UK.

GELDERS, L. F. and **VAN WASSENHOVE, L. N.**, 1985, Capacity Planning in MRP, JIT and OPT: A Critique, *Engineering Costs and Production Economics*, **9**, 201-209

GODIN, V. B., 1978, Interactive Scheduling: Historical Survey and State of the Art, *AIIE Transactions*, **9**, 331-337

GOODWIN, J. S. and **GOODWIN, J. C.**, 1982, Operating Policies for Scheduling Assembled Products, *Decision Sciences*, **13**, 585-603.

GOODWIN, J. S. and **WEEKS, J. K.**, 1986, Evaluating Scheduling Policies in a Multi-Level Assembly System, *International Journal of Production Research*, **26**, 247-257.

GOLDRATT, E. M. and **COX, J.**, 1985, *The Goal*, (Croton-on-Hudson, NY: North

River Press Inc.)

GRAVES, S. C., 1981, A Review of Production Scheduling, *Operations Research*, 29, 646-675.

HALL, R. W., 1981, Driving the Productivity Machine - Production Planning and Control in Japan, *American Production & Inventory Control Society Inc.*, Falls Church, Virginia.

HARDGRAVE W. W. and **NEMHAUSER, G. L.**, 1963, A Geometric Model and a Graphic Algorithm for a Sequencing Problem, *Operations Research*, 11, 889-900.

HASTINGS, N. A. J., MARSHALL, P.H. and **WILLIS, R. J.**, 1982, Schedule based MRP: An Integrated Approach to Production Scheduling and Materials Requirements Planning, *Journal of the Operational Research Society*, 33, 1021-1029.

HEUSER, W. A. Jr. and **WYNNE, B. E. Jr.**, 1963, An Application of the Critical Path Method to Job Shop Scheduling - A Case Study, *Management Technology*, 3, 128-144

HOGG, R. V. and **LEDOLTER, J.**, 1987, *Engineering Statistics*, (New York: Collier Macmillan)

HUANG, P. Y., 1984, A Comparative Study of Priority Dispatching Rules in a Hybrid Assembly Job Shop, *International Journal of Production Research*, 22, 375-387.

IGNALL, E. J. and **SCHRAGE, L. E.** 1965, Application of the Branch and Bound Technique to Some Flow-Shop Scheduling Problems, *Operations Research*, 13, 400-412

INGLESBY, A., 1991, MRP? Finite Scheduling? Yes, No & Maybe, *Manufacturing Systems*, March, 57-60

JOHNSON, S. M. 1954, Optimal two- and three-stage Production Schedules with Setups Included, *Naval Research Logistics Quartely*, 1, 61-68

KELLEY, J. and **WALKER, M.**, 1959, Critical Path Planning and Scheduling, *Proceedings of the Eastern and Joint Computer Conference*, Boston.

KEMPF, K. G., 1989, Manufacturing Planning and Scheduling: where we are and where we need to be, *Proceedings of the 5th Conference on Artificial Intelligence Applications*, Miami, Fl, March, 6-10.

KING, J. R., 1976, The Theory-Practice Gap in Job-Shop Scheduling, *Production Engineer*, March, 137-143

KING, J. R., 1985, *Management of Engineering Production*, (London: Frances Pinter).

MALCOLM, D., ROSEBOOM, J., CLARK, C. and **FAZAR, W.**, 1959, Application of a Technique for Research and Development Program Evaluation, *Operations Research*, 7, 646-669

MAXWELL, W. L., 1969, Priority Dispatching and Assembly Operations in a Job Shop, *RAND Corporation Memorandum RM-5370-PR.*

MAXWELL, W. L. and **MEHRA, M.**, 1968, Multi-Factor Rules for Sequencing with Assembly Constraints, *Naval research Logistcs Quartely*, 15, 241-254

MILLER, R. W., 1962, How to Plan and Control with PERT, *Harvard Review Magazine*, 40, 93-104

NAJI, Z., 1991, *Correcting for Data Imprecision in MRP2 Systems*, Unpublished Ph.D. Dissertation, Cranfield Institute of Technology, U.K.

OHNO, Taiichi, 1988, *Toyota Production System*, (Cambridge, Ma: Productivity Press).

ORLICKY, J. A., 1975, *Materials Requirements Planning*, (New York, McGraw-Hill)

PANWALKER, S. S. and **ISKANDER, W.**, 1977, A Survey of Scheduling Rules, *Operations Research*, 25, 45-61

PHILIPOOM, P.R., RUSSEL, R. S. and **FRY, T. D.**, 1991, A Preliminary Investigation of Multi-Atributte Based Sequencing Rules for Assembly Shops, *International Journal of Production Research*, 29, 739-754.

POCOCK, J. W., 1962, PERT as an Analytical Aid for Program Planning - Its Payoff and Problems, *Operations Research*, 10, 893-903

ROCHETTE, R. and **SADOWSKI, R. P.**, 1976, A Statistical Comparison of the Performance of Simple Dispatching Rules for a Particular Set of Job Shops, *International Journal of Production Research*, 14, 63-75

RODAMMER, F. A. and **WHITE, K. P.**, 1988, A recent Survey of Production

Scheduling, *IEEE Transactions on Systems, Man and Cybernetics,* 18,, 841-851

RUSSEL, R. S. and **TAYLOR III, B. W.**, 1985a, An Evaluation of Scheduling Policies in a Dual Resource Constrained Assembly Shop, *IIE Transactions*, 17, 219-232

RUSSEL, R. S. and **TAYLOR III, B. W.**, 1985b, An Evaluation of Sequencing Rules for an Assembly Shop, *Decision Science*, 16, 196-212

SARIN, S. C. and **DAS, S. K.**, 1988, Production Control Implications for CIMS, *IEEE Transactions*, 343-349

SAVAGE, E. and **MIKURAK, M.**, 1985, The integration of a Finite Scheduling System into Manufacturing Resource Planning, *AUTOFACT' 85 Conference Proceedings,* Detroit, USA, 35-44

SCULLI, D., 1980, Priority Dispatching Rules in Job Shops with Assembly Operations and Random Delays, *Omega*, 8, 227-234

SHERIDAN, J. H., 1989, Preaching the Gospel of Finite Scheduling, *Industry Week*, August 21, 42-43

SIEGEL, G. B., 1971, *An Investigation of Job Shop Scheduling for Jobs with Assembly Constraints*, unpublished Ph.D. dissertation, Cornell University, USA

SPACHIS, A. S. and **KING, J. R.**, 1979, Job-Shop Scheduling Heuristics with Local Neighbourhood Search, *International Journal of Production Research*, 17, 507-526

TURNER, J., 1991, Finite Scheduling: MRP-II's final frontier, *Manufacturing Systems*, February, 62-64

WAGNER, H. M., 1969, *Principles of Operations Research*, (New Jersey, Prentice-Hall inc.)

WOODGATE, H. S., 1989, MRP-III - Material Flow Control in JIT Manufacturing, *Autofact' 89, Conference Proceedings*, October 30/November 2, Detroit, Michigan, 9-24

WU, B., 1992, *Manufacturing Systems Design and Analysis*, (London: Chapman and Hall)

APPENDIX 1 - SOME RELEVANT SEQUENCING RULES FOR ASSEMBLY SHOP

i) From Maxwell's Study **(Maxwell, 1967)**

MAXRWD-NR	The priority of an operation in queue equals the difference between the maximum remaining work over the branches of the job and the remaining work on the branch for this operation. This difference is computed ate the time the operation enters queue and is not re-evaluated if work is completed on operations of other branches of the job.
MAXNRD-SPT	The priority of an operation in queue equals the difference between the maximum number of remaining operations over the branches of the job and the number of operations remaining on the branch on the operation. This difference is computed for each operation in queue every time an operation is to be selected from queue. (SPT is used to break ties).
NUB-SPT	Select the operation whose parent job has the smallest number of uncompleted parts, with ties broken by the shortest processing time. The shortest processing time is also used for selecting operations in the assembly centre.

ii) From Sculli's study **(Sculli, 1980)**

FLOAT-SPT	Select the operation with the smallest float. The float is computed by performing a critical path analysis on the remaining operations network and putting the latest possible job completion time equal to the earliest possible completion time. Priority is re-evaluated each time an operation is selected. The shortest processing time is used as a tie-

	breaker, and also for selecting operations in the assembly centre.
SLACK	Select the operation with the smallest slack. The slack is computed by performing a critical path analysis on the remaining operations sequence network and using the due-date as the latest possible completion time.

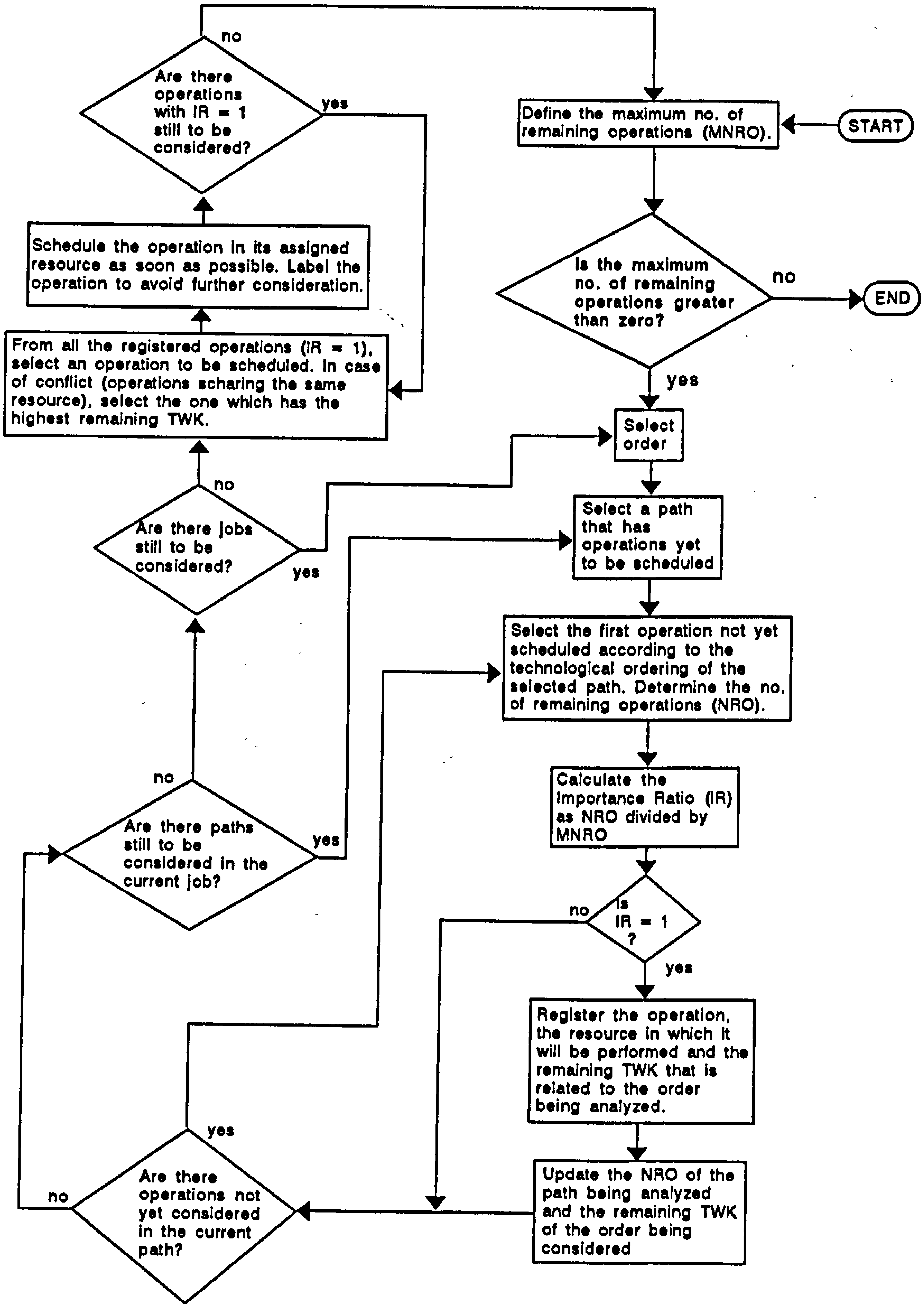
iii) From **Russel and Taylor (1985a, b)**

BS + ROPT^2	Select the operation which has the smallest branch slack and remaining number of operations squared. This combined rule paces the completion of items in a common assembly and coordinates all items of a job by setting branch due-dates that accumulate to the job due-date.
LP + ROPT^2	This rule processes items first that have the longest sum of processing times from component to parent to final product completion. Rankings within a job vary, but are coordinated to a degree by a common ROPT^2 factor.

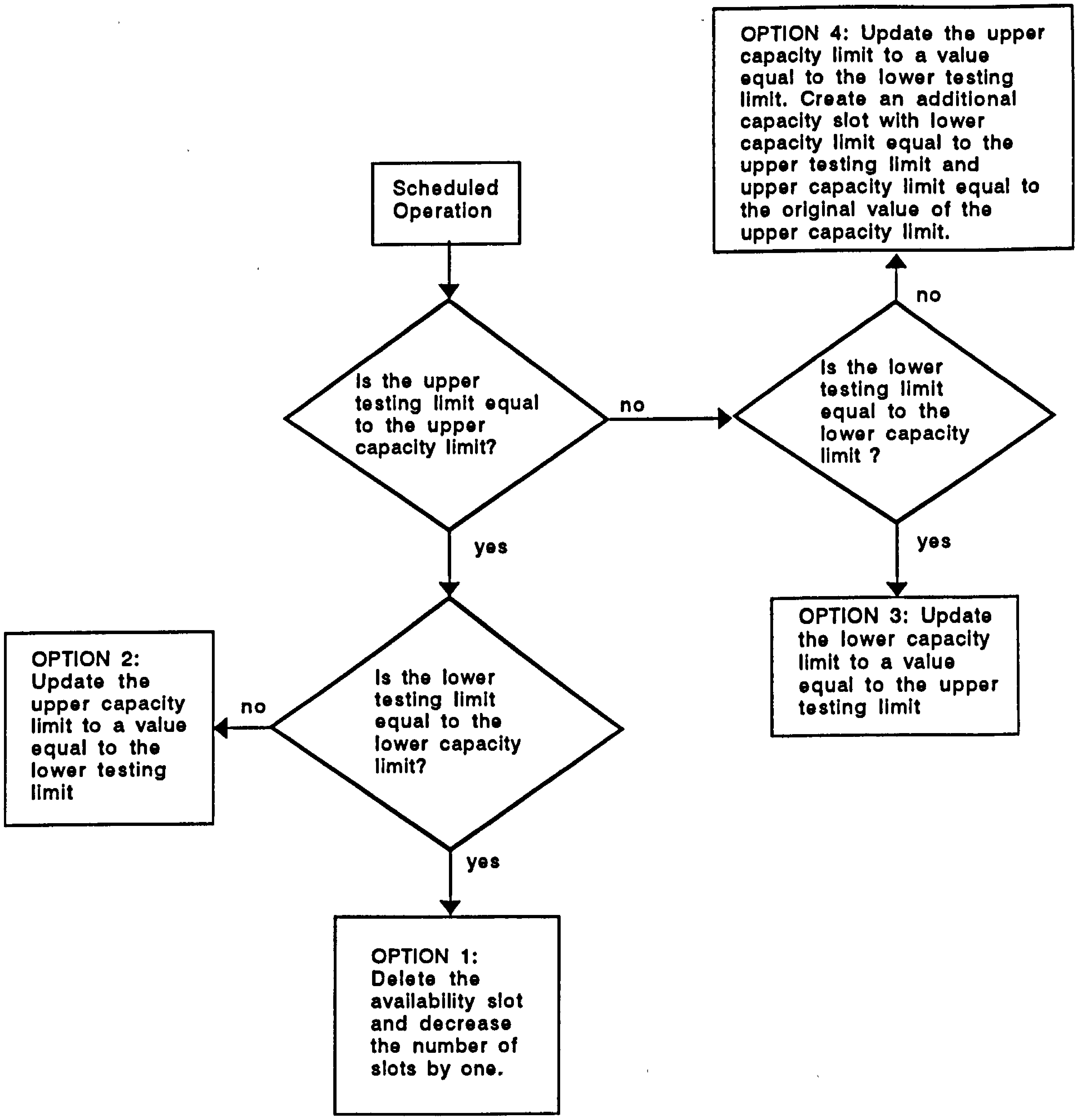
iv) From **Fry** ***et. al.,*** **(1989)**

LVLSPT	This rule assigns priority to an operation based on its level in the BOM. An operation located near the top of the BOM receives priority over an operation located in a lower level. All ties among operations are broken by the SPT rule.

Importance Ratio – Total Work Content



The Updating Module



THE ALGORITHM

Step 1. This step is related to updating the Resource Capacity array [A]. As the scheduling process develops, available capacity slots are being progressively occupied, thus altering the number of slots and their limits. If $V=A_{k3}$ go to step 1.1, otherwise go to step 1.2.

Step 1.1. OPTION 1: If $W=A_{k2}$, the required capacity matches exactly the capacity available. Thus, this availability slot is deleted from the array and the number of elements in the array is updated by decreasing n_k by one.

OPTION 2: On the other hand, if $W\neq A_{k2}$, the number of slots is not affected and the new capacity limit is given by:

$$A_{k3}=W$$

See details in the next page. Go to step 1.3.

Step 1.2. OPTION 3: If $V<A_{k3}$ and $W=A_{k2}$, no alteration on the number of slots is necessary to be done. The lower capacity limit is updated as follows:

$$A_{k2}=V$$

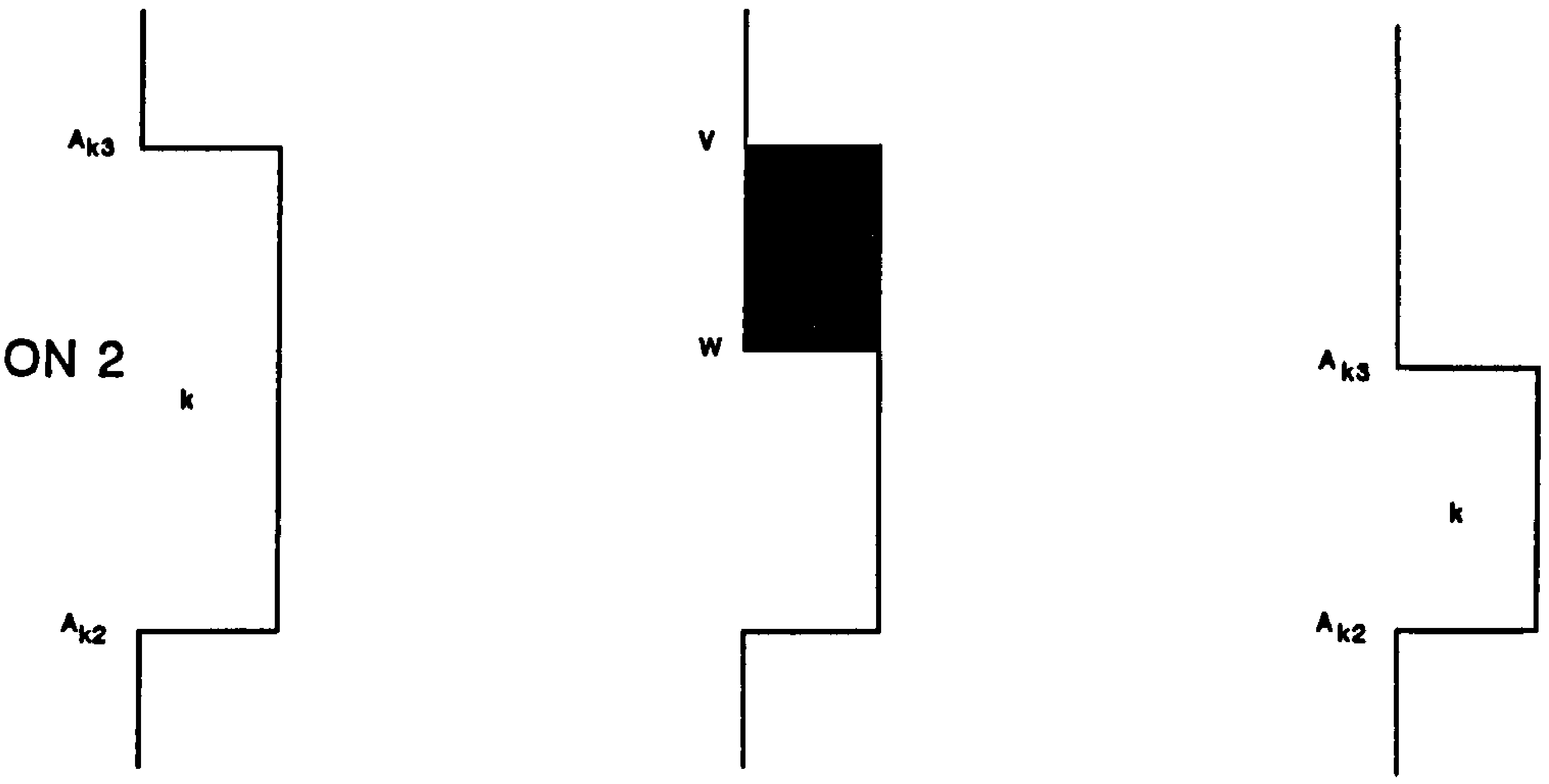
OPTION 4: However, if $W>A_{k2}$, then it is necessary to add a new availability slot.

$$\begin{aligned}
&V^{*}=A_{k3}\\
&A_{k3}=W\\
&A_{n_k+1,1}=A_{k1}\\
&A_{n_k+1,2}=V\\
&A_{n_k+1,3}=V^{*}
\end{aligned}$$

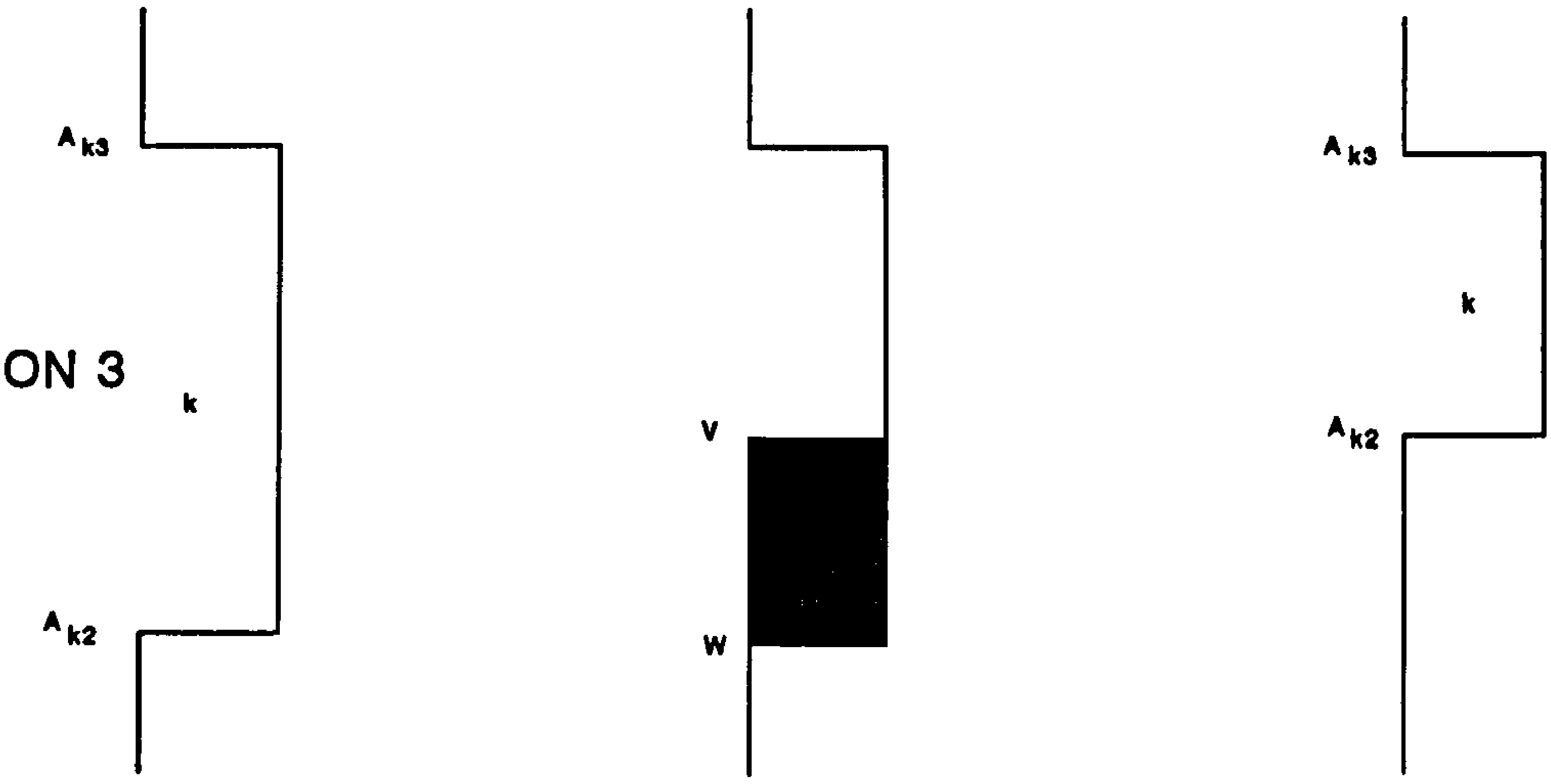
Increase n_k by one and go to step 1.3.

Step 1.3. This step sorts the updated array Resource capacity according to the type of loading.
i) If the scheduling process was done in forward way then the capacity array is sorted in ascending way. The justification is due to the fact that in the forward way all searches on this array are done sequentially from the lowest to the highest availability index.
ii) On the other hand, if the scheduling was done in backward way, then the array Resource Capacity is sorted in descending manner. The sorting of the capacity array in descending manner facilitates the searching process when in the backward manner.

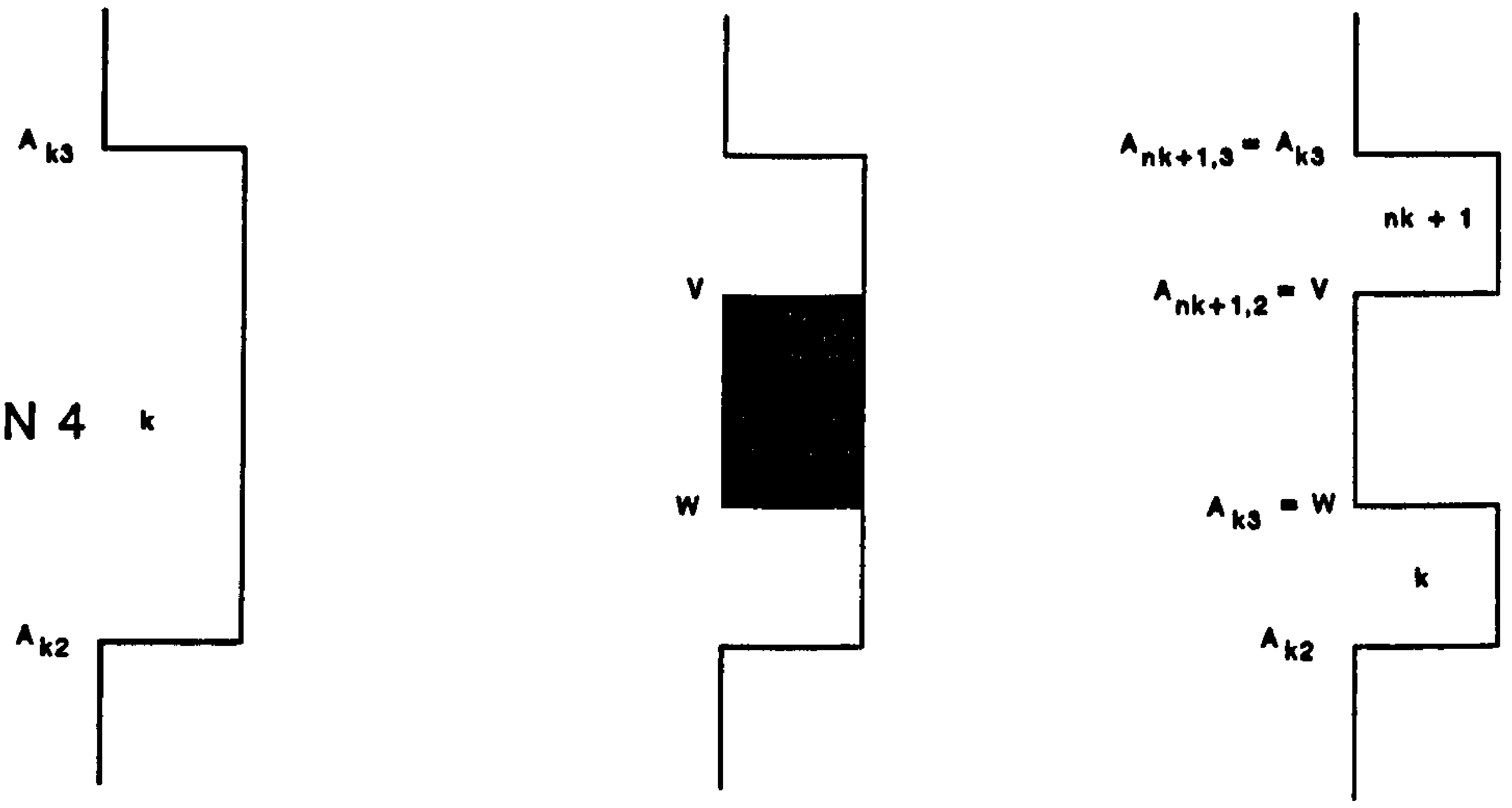
OPTION 2



OPTION 3

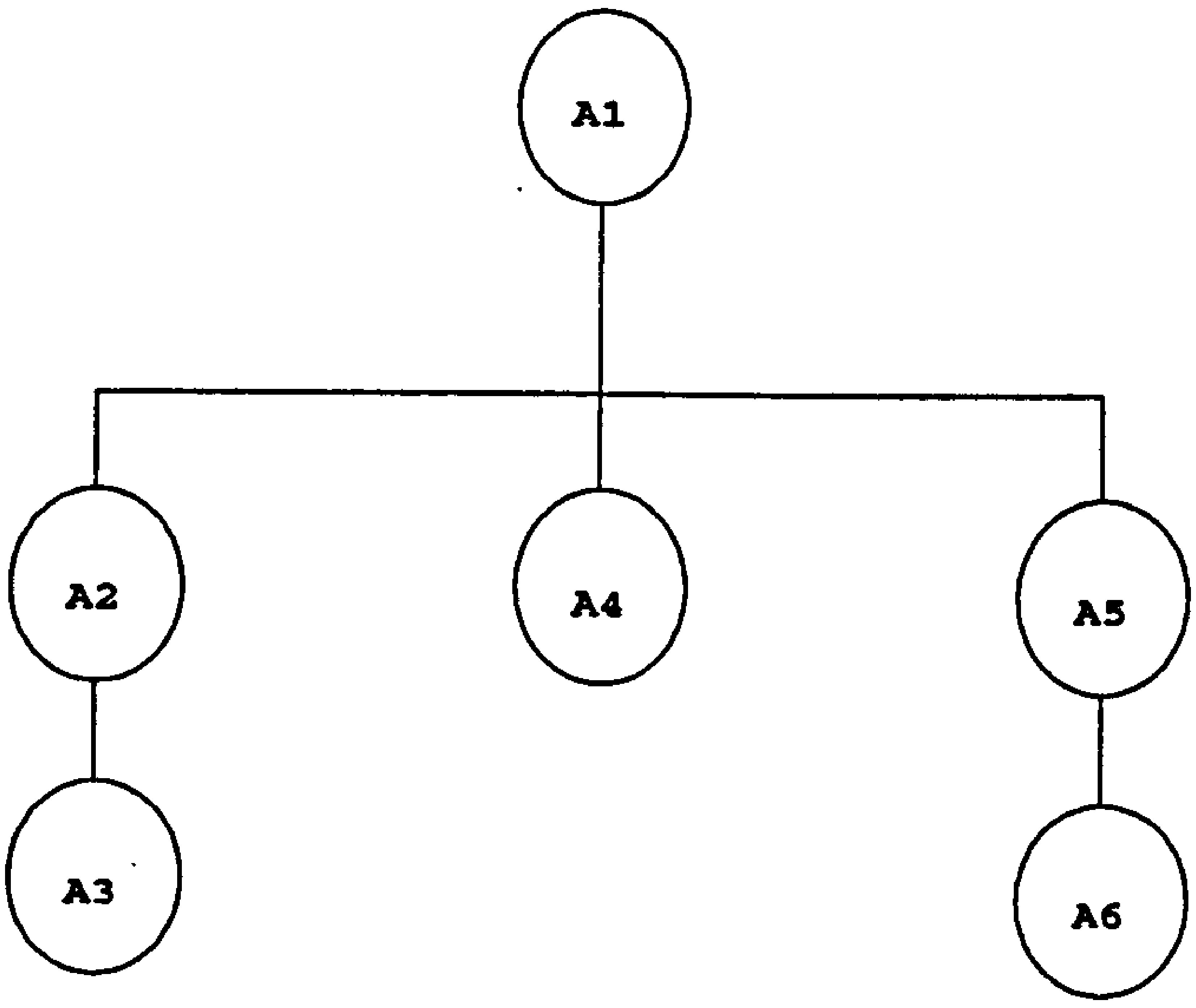


OPTION 4

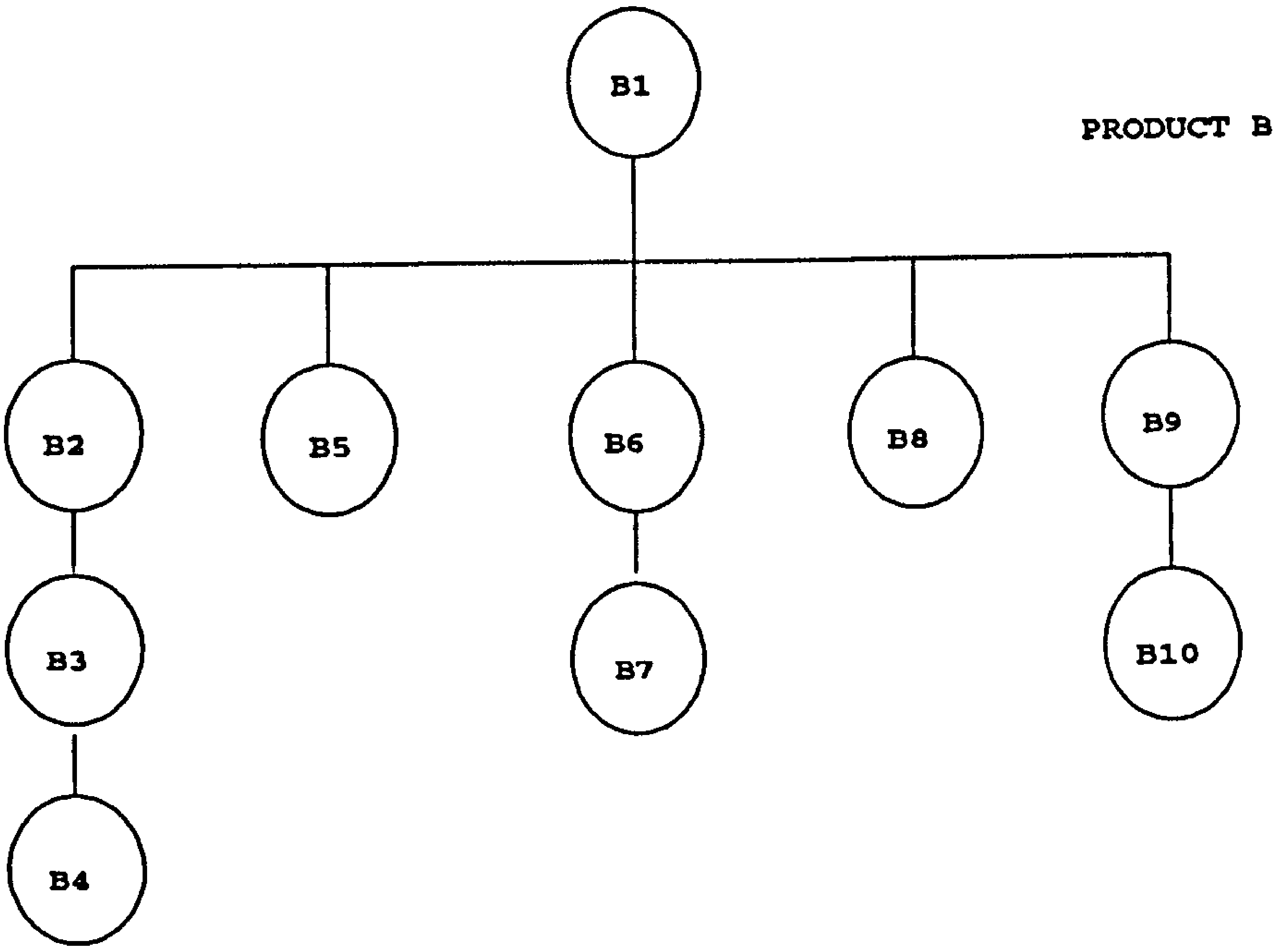


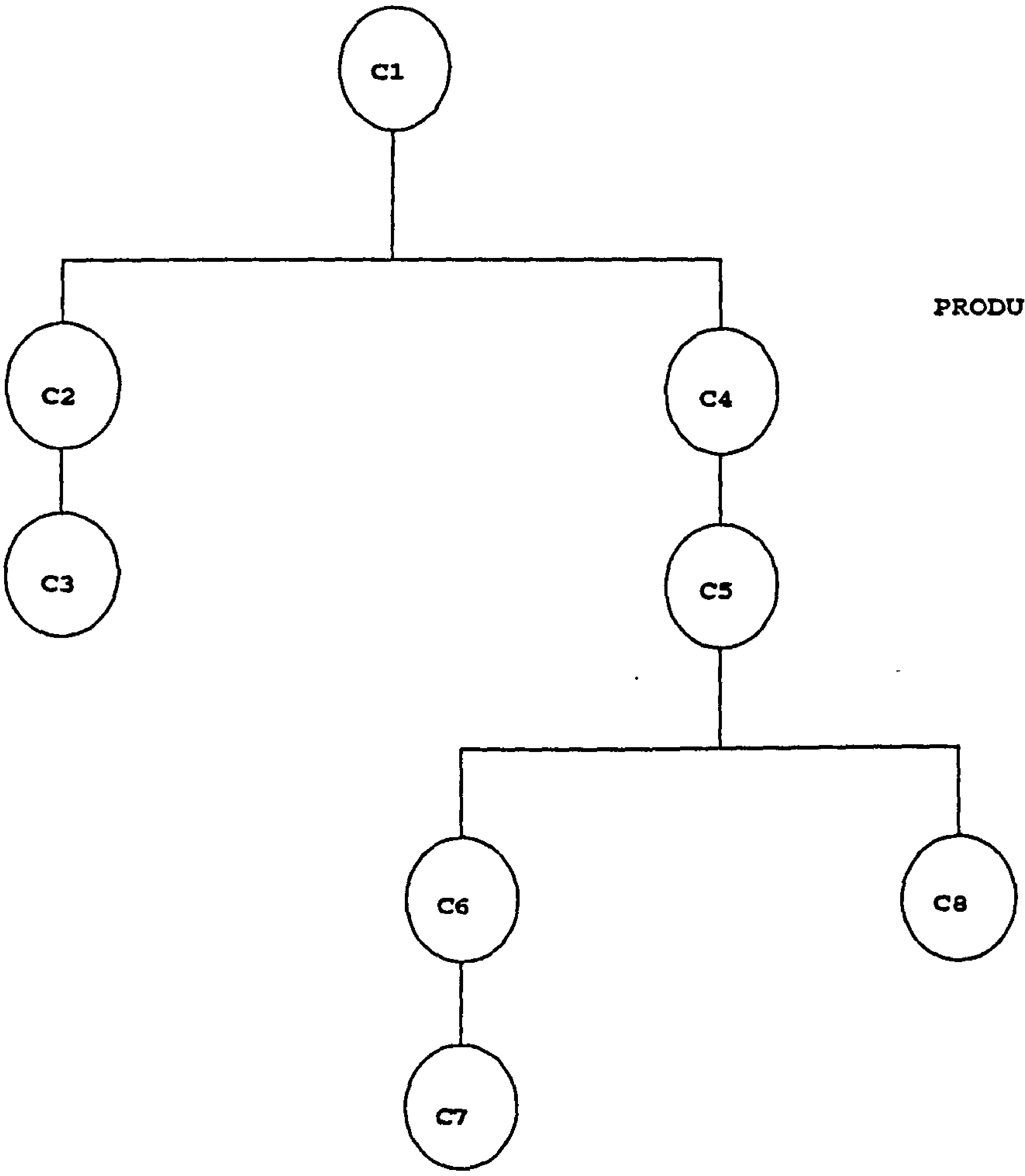
APPENDIX 4: THE TESTED BILL OF MATERIALS (BOMs)

FLAT STRUCTURE



PRODUCT A

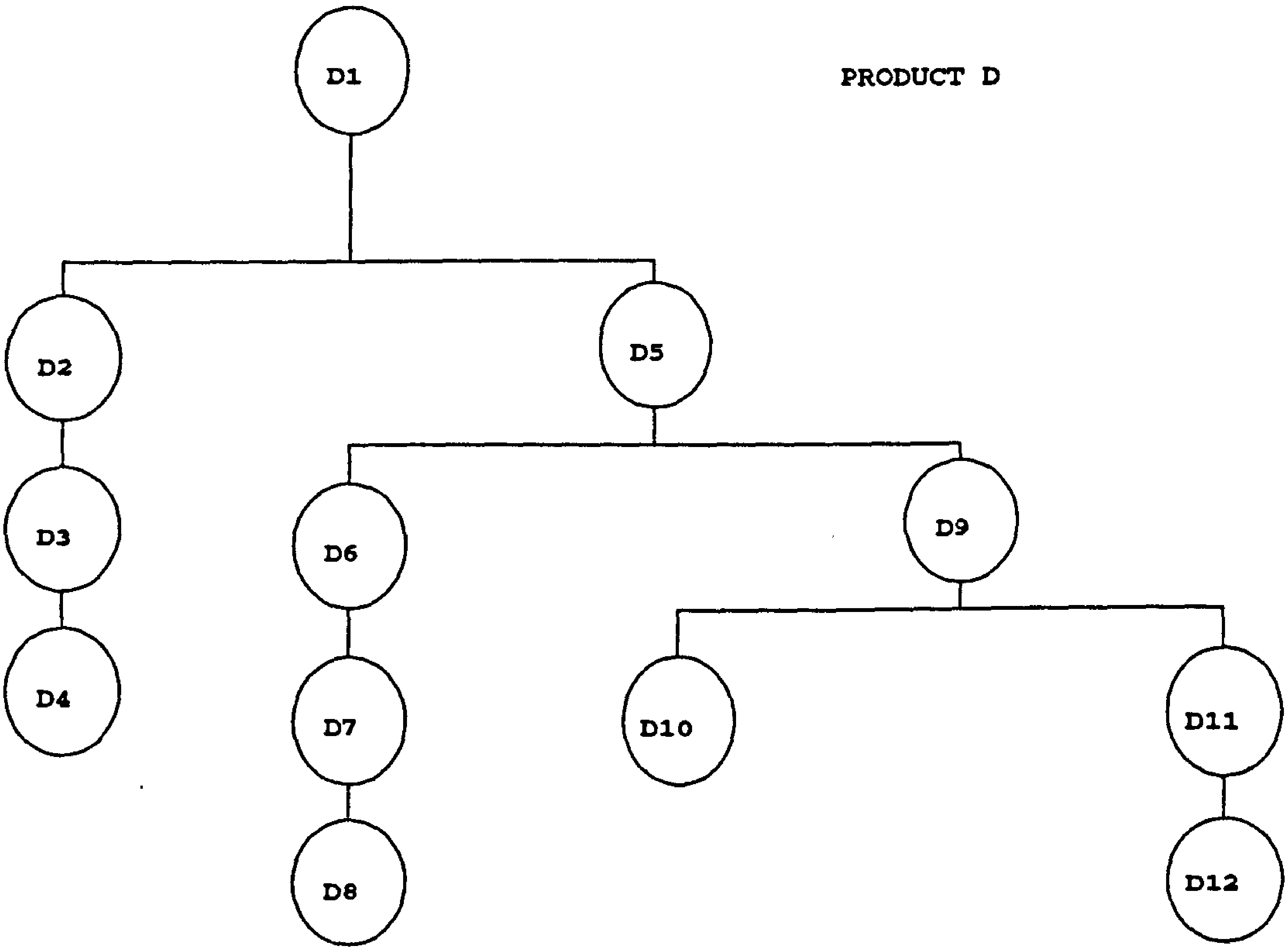

B1
PRODUCT B
B2
B5
B6
B8
B9
B3
B7
B10
B4



TALL STRUCTURE

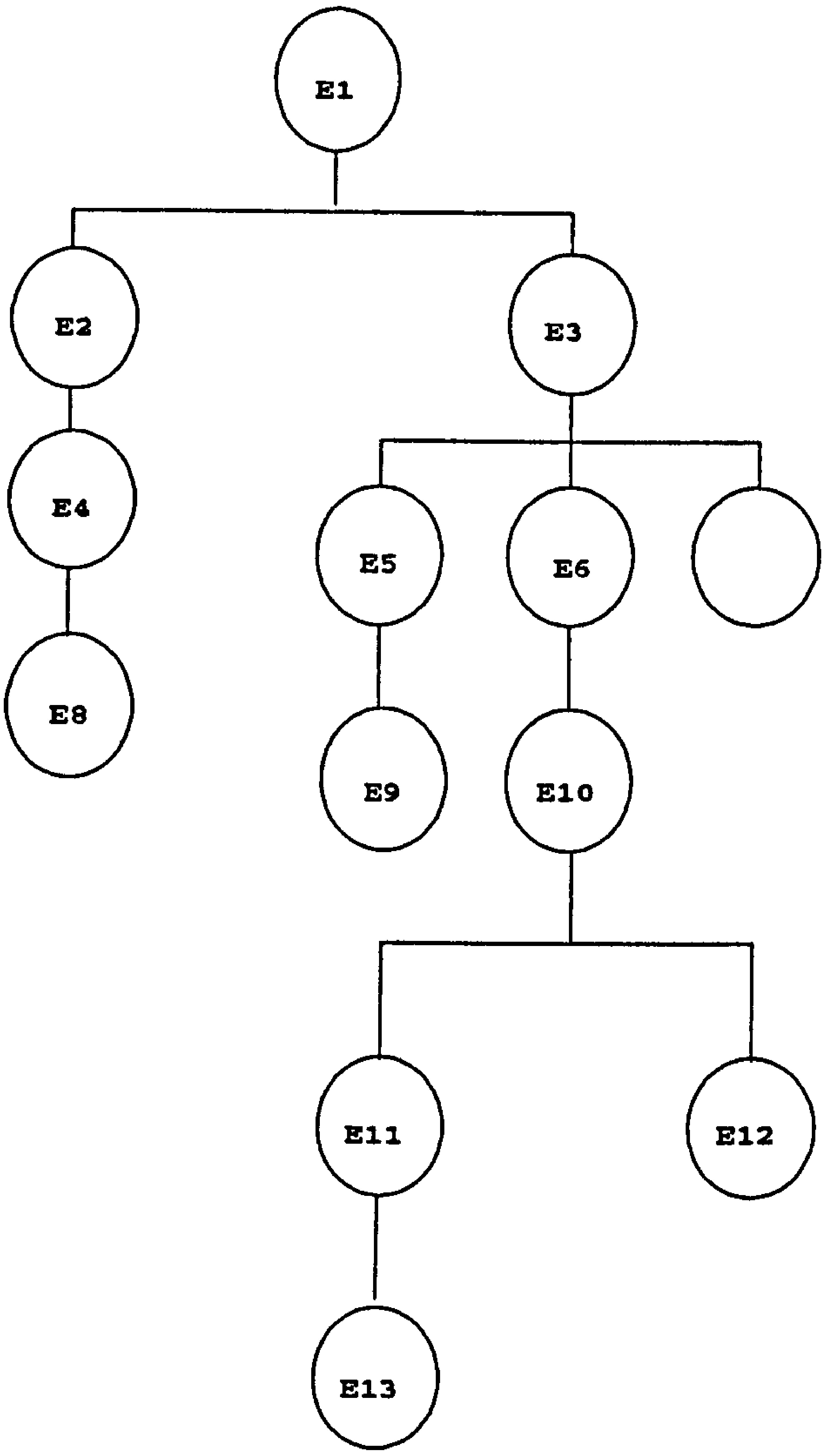
PRODUCT C

FLAT STRUCTURE

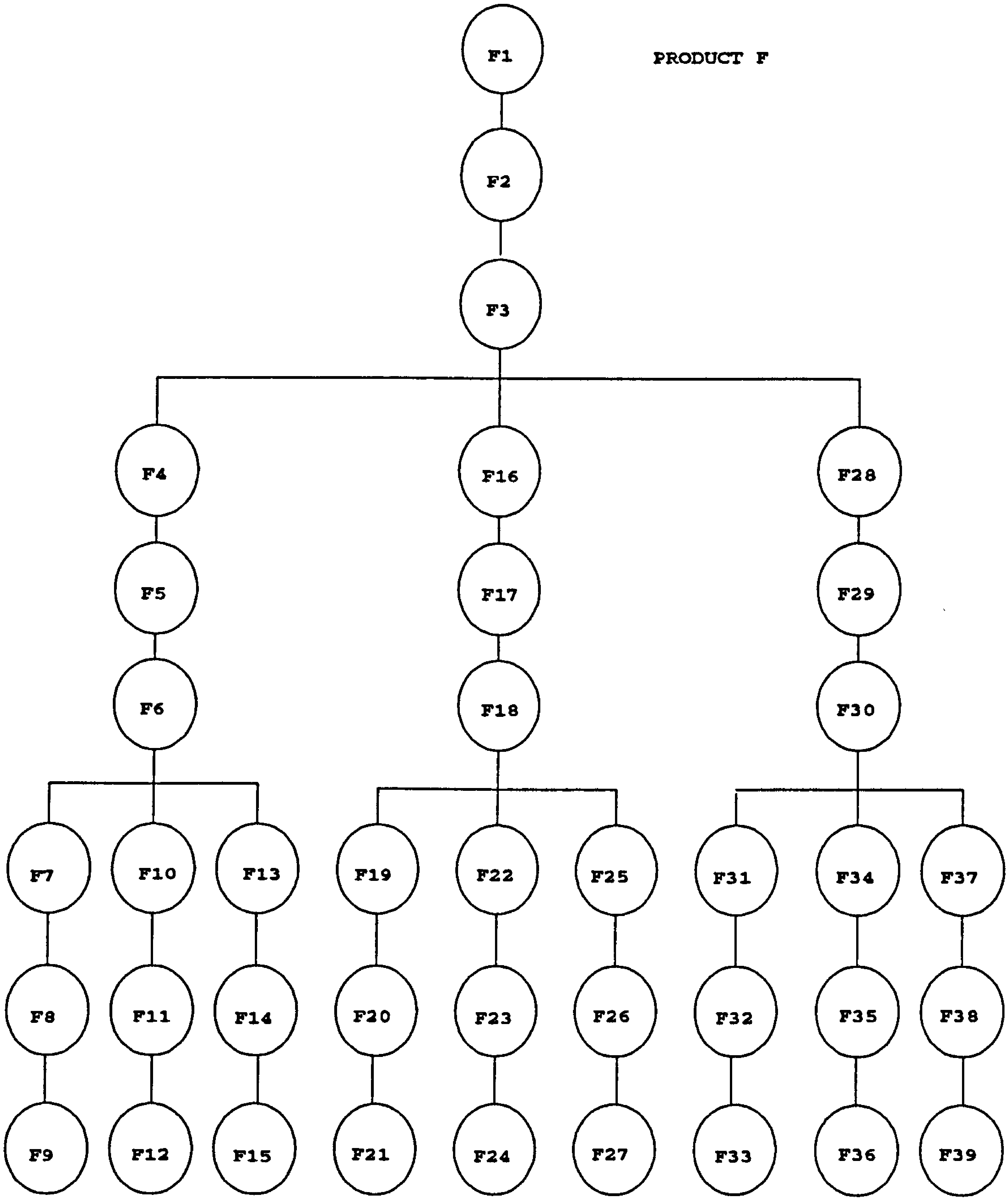


MIXED STRUCTURE

PRODUCT E

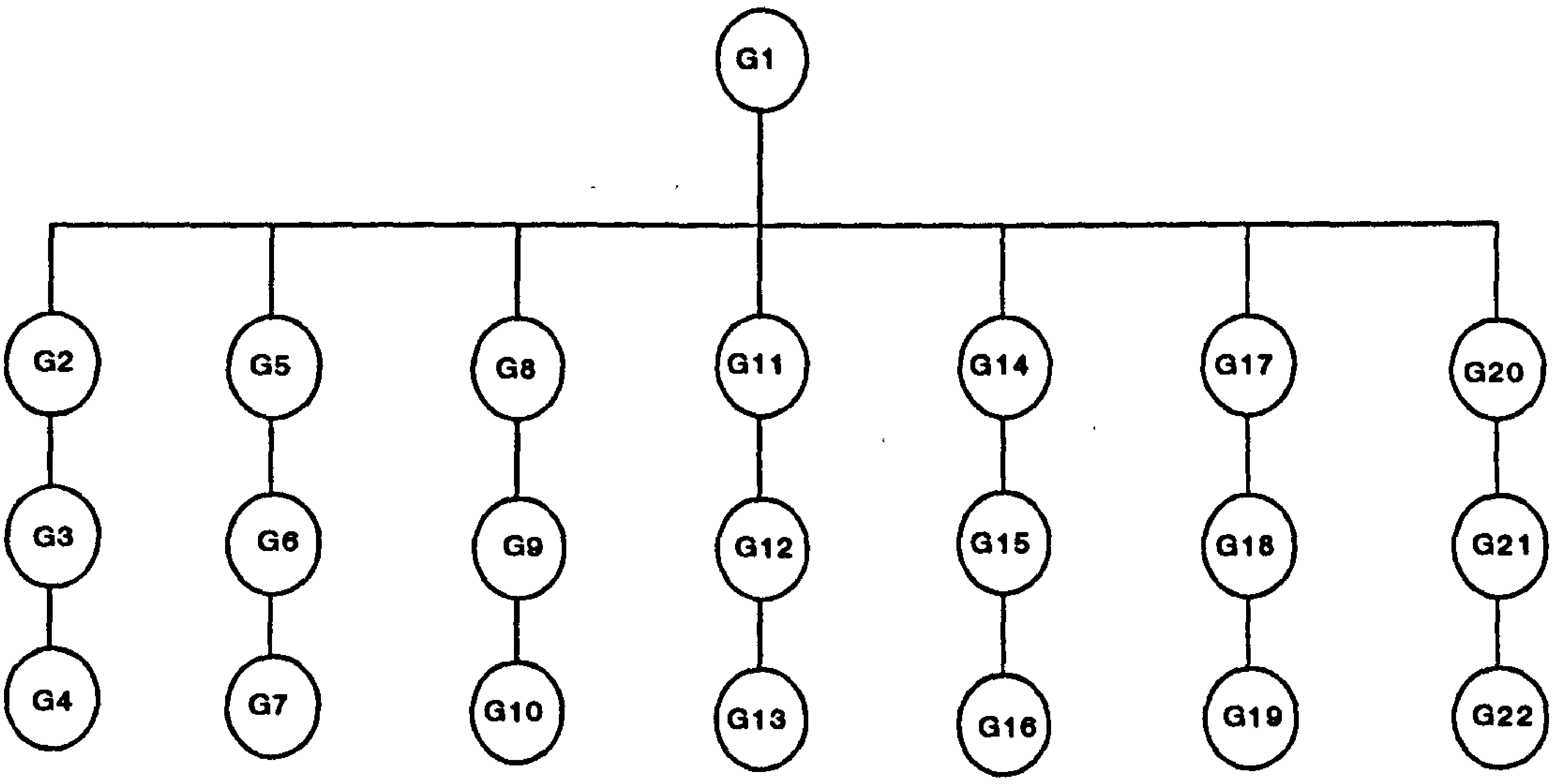


MIXED STRUCTURE

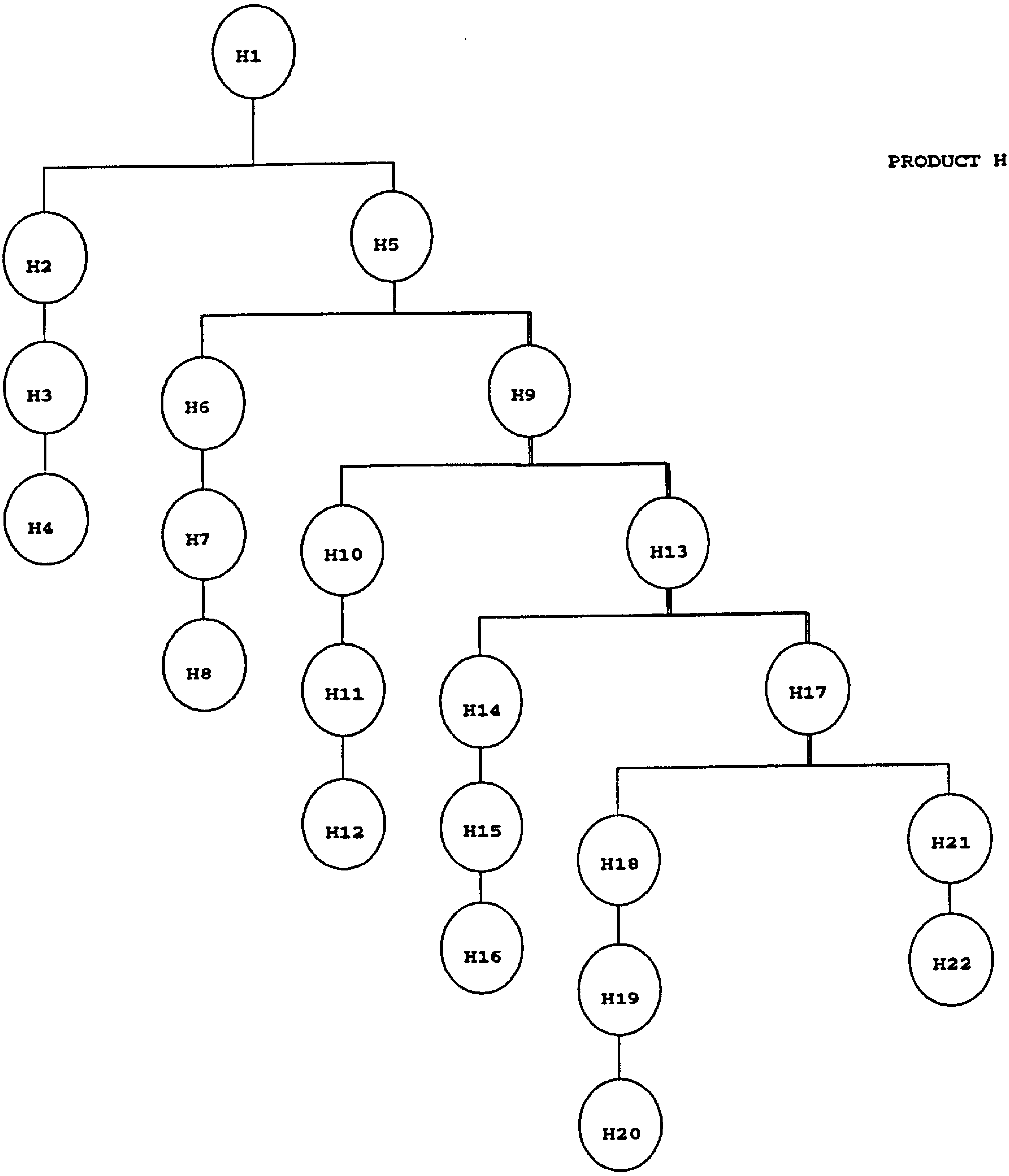


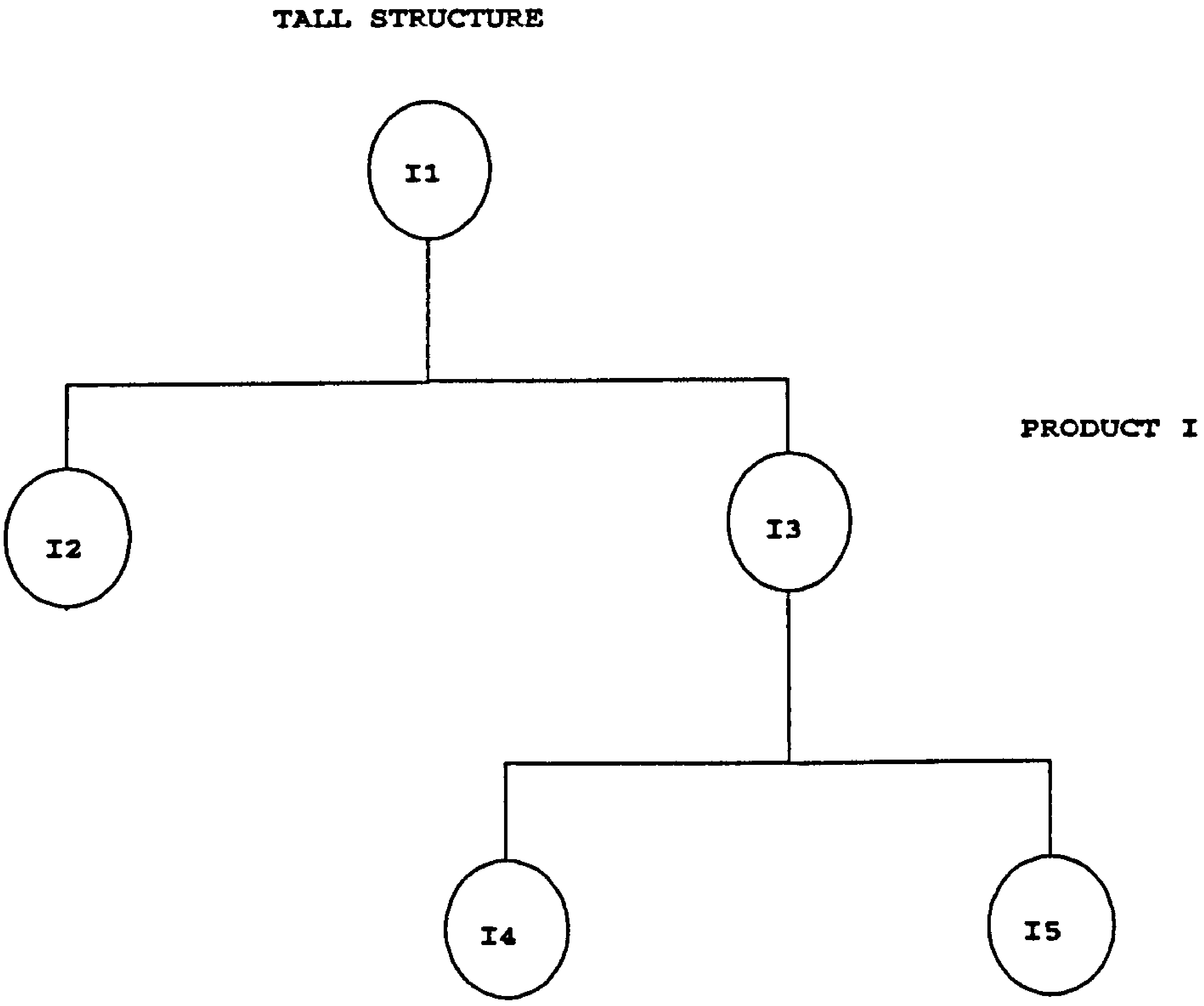
FLAT STRUCTURE

PRODUCT G

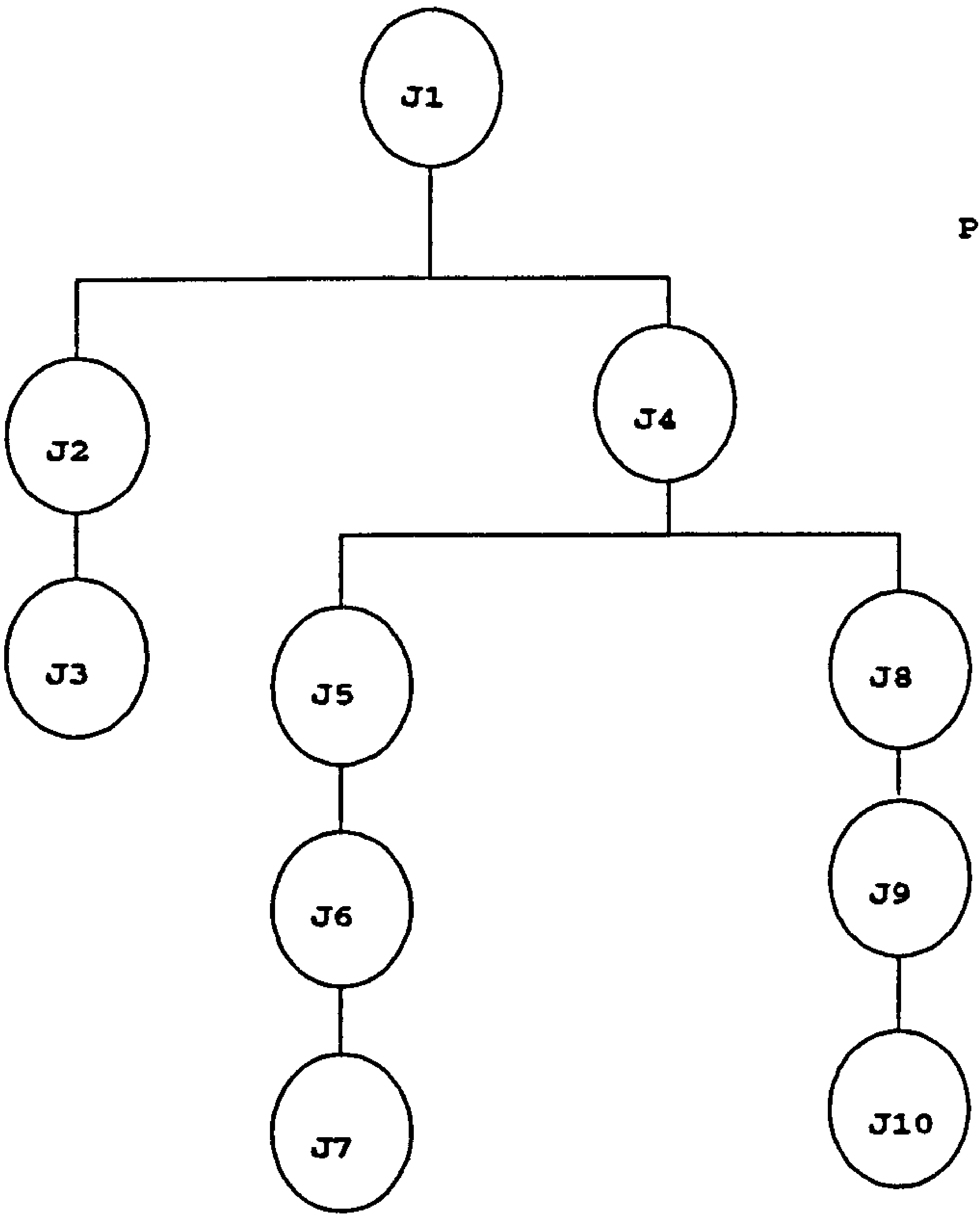


TALL STRUCTURE



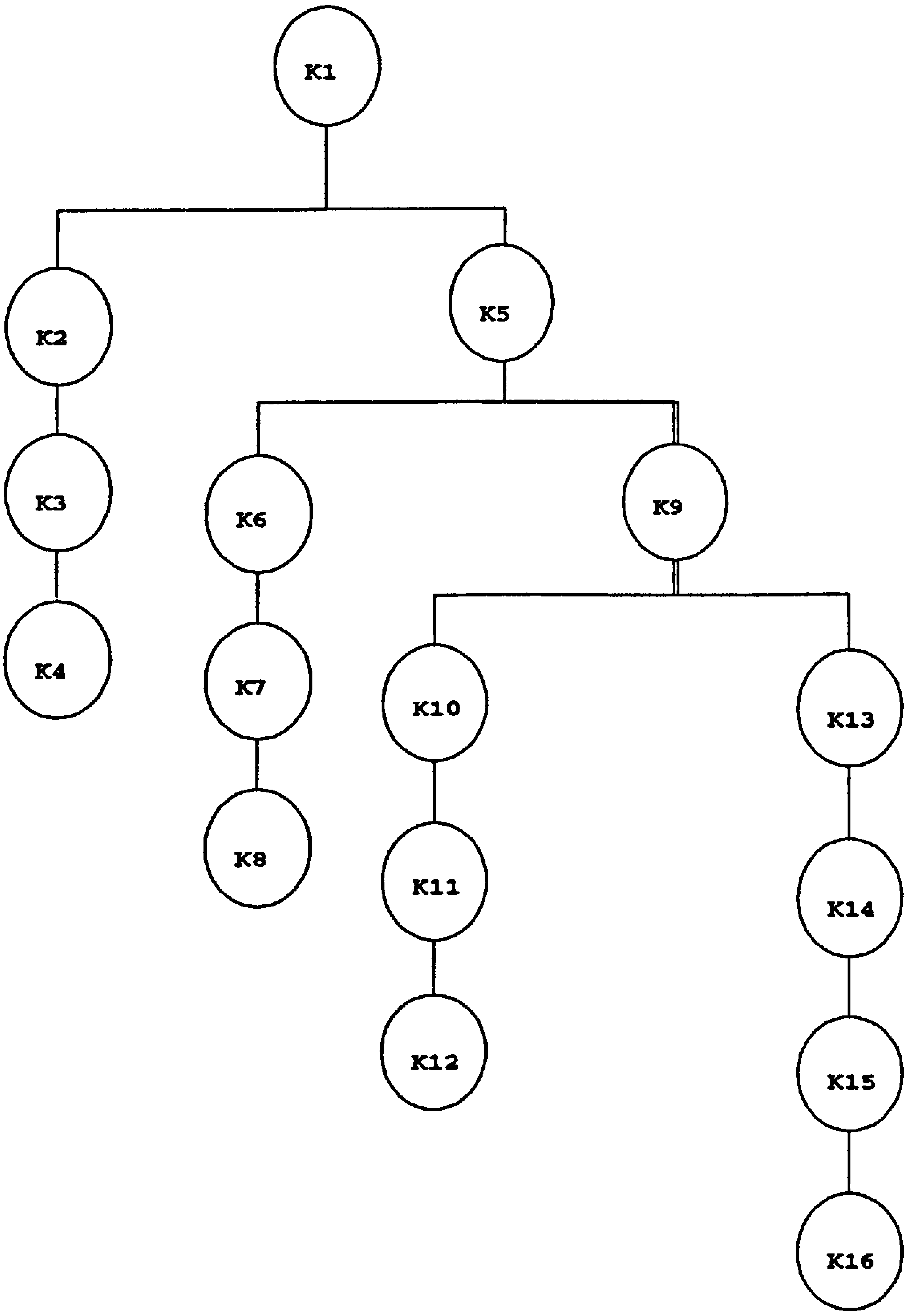

TALL STRUCTURE
I1
PRODUCT I
I2
I3
I4
I5

TALL STRUCTURE



PRODUCT J

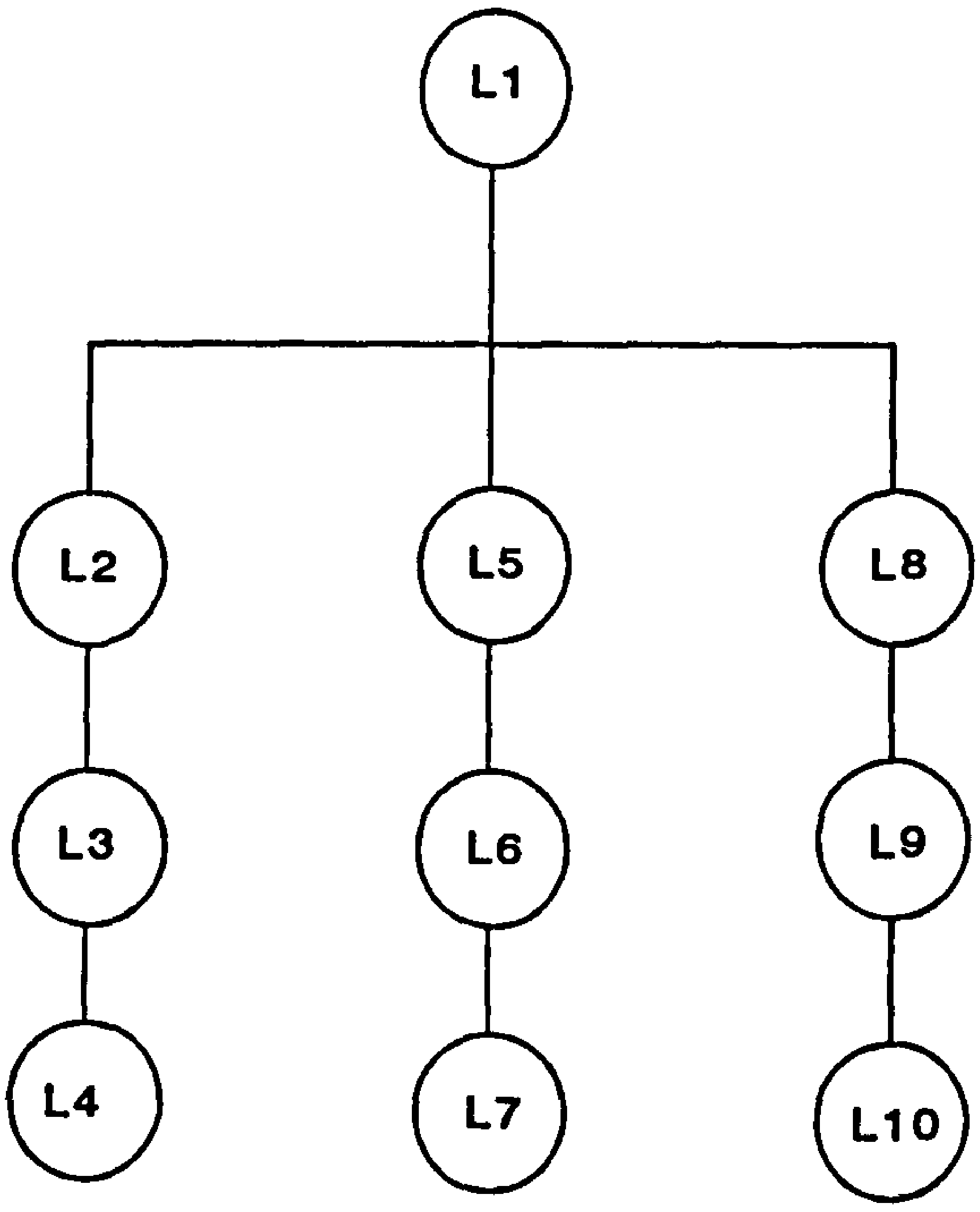
TALL STRUCTURE



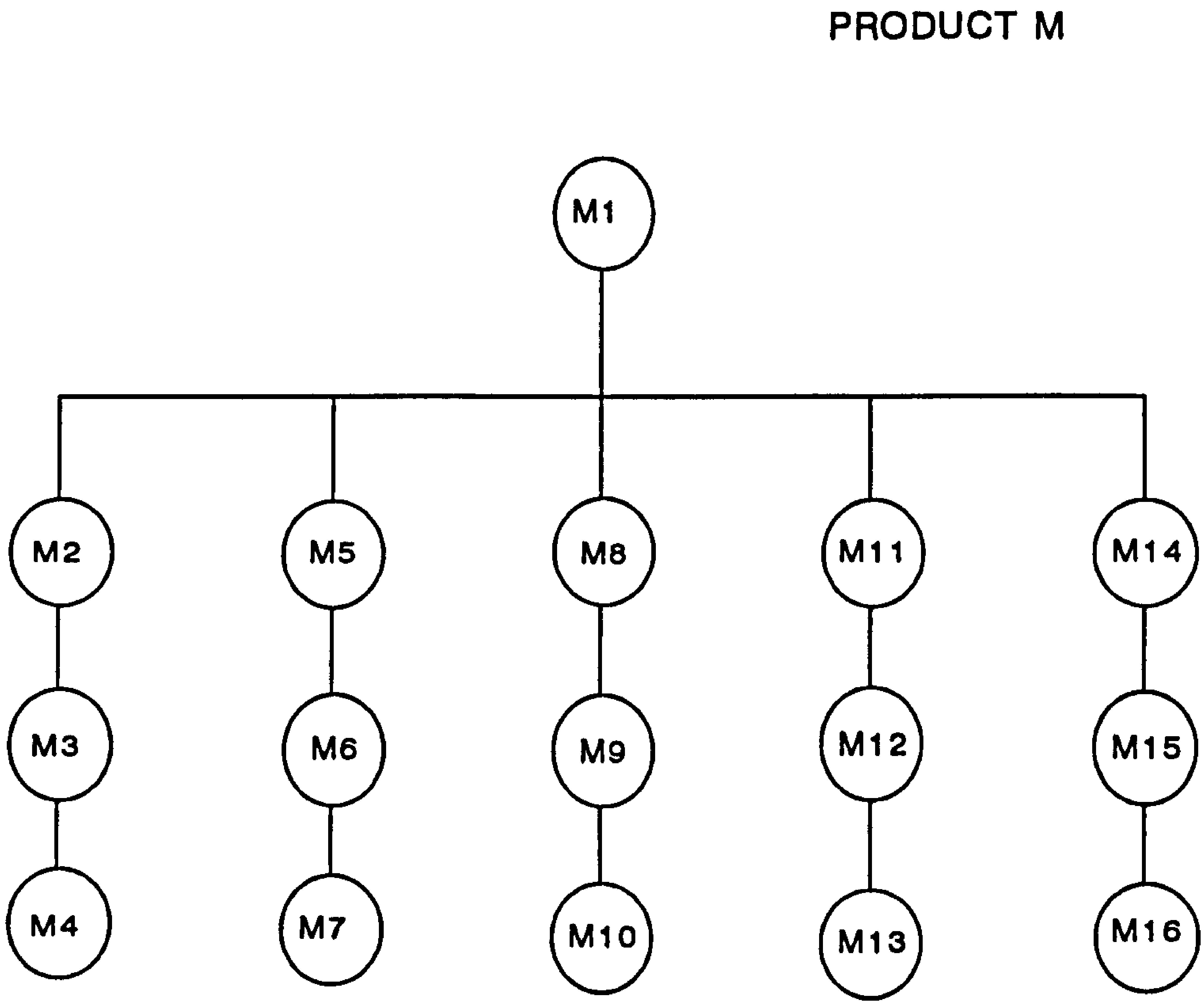
PRODUCT K

FLAT STRUCTURE

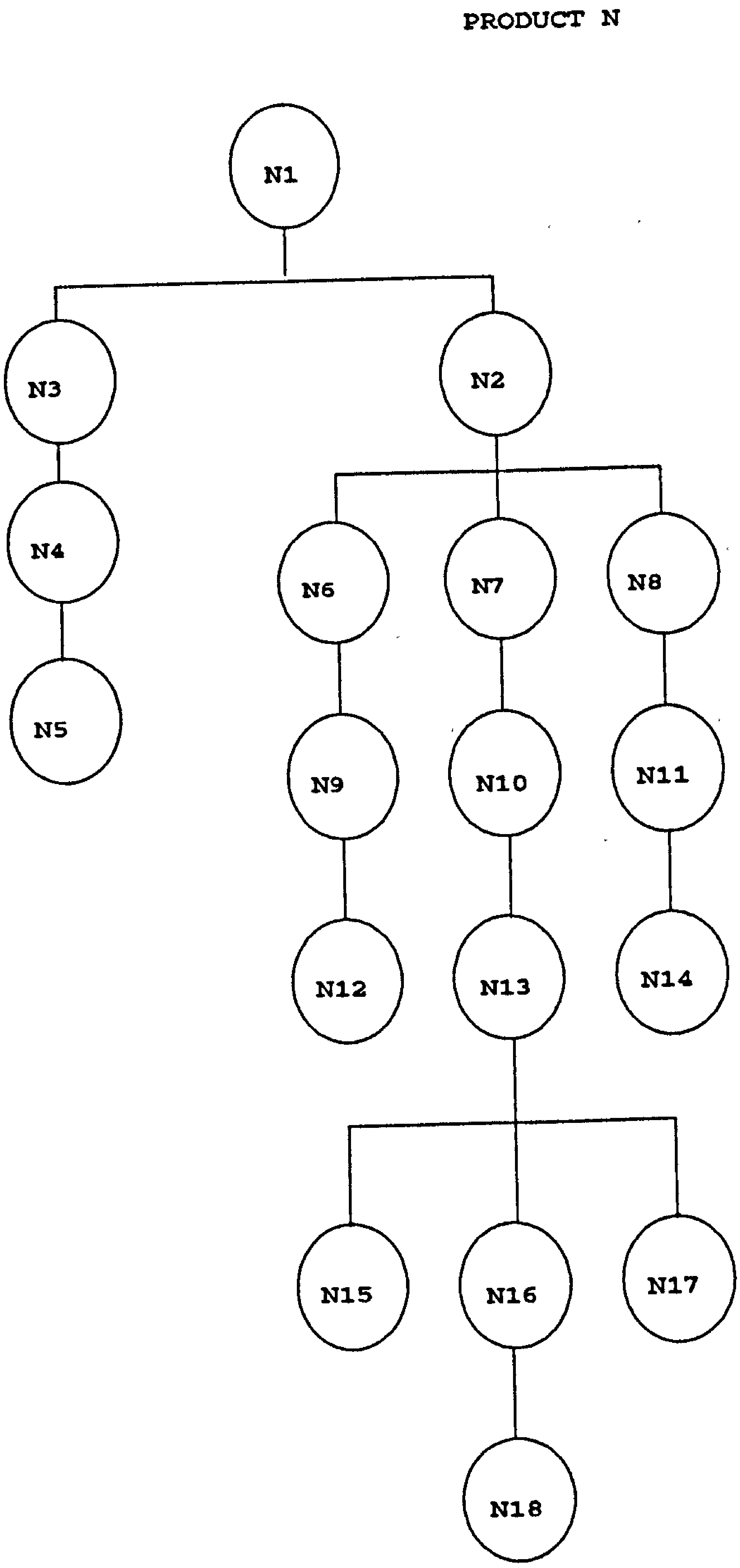
PRODUCT L



FLAT STRUCTURE

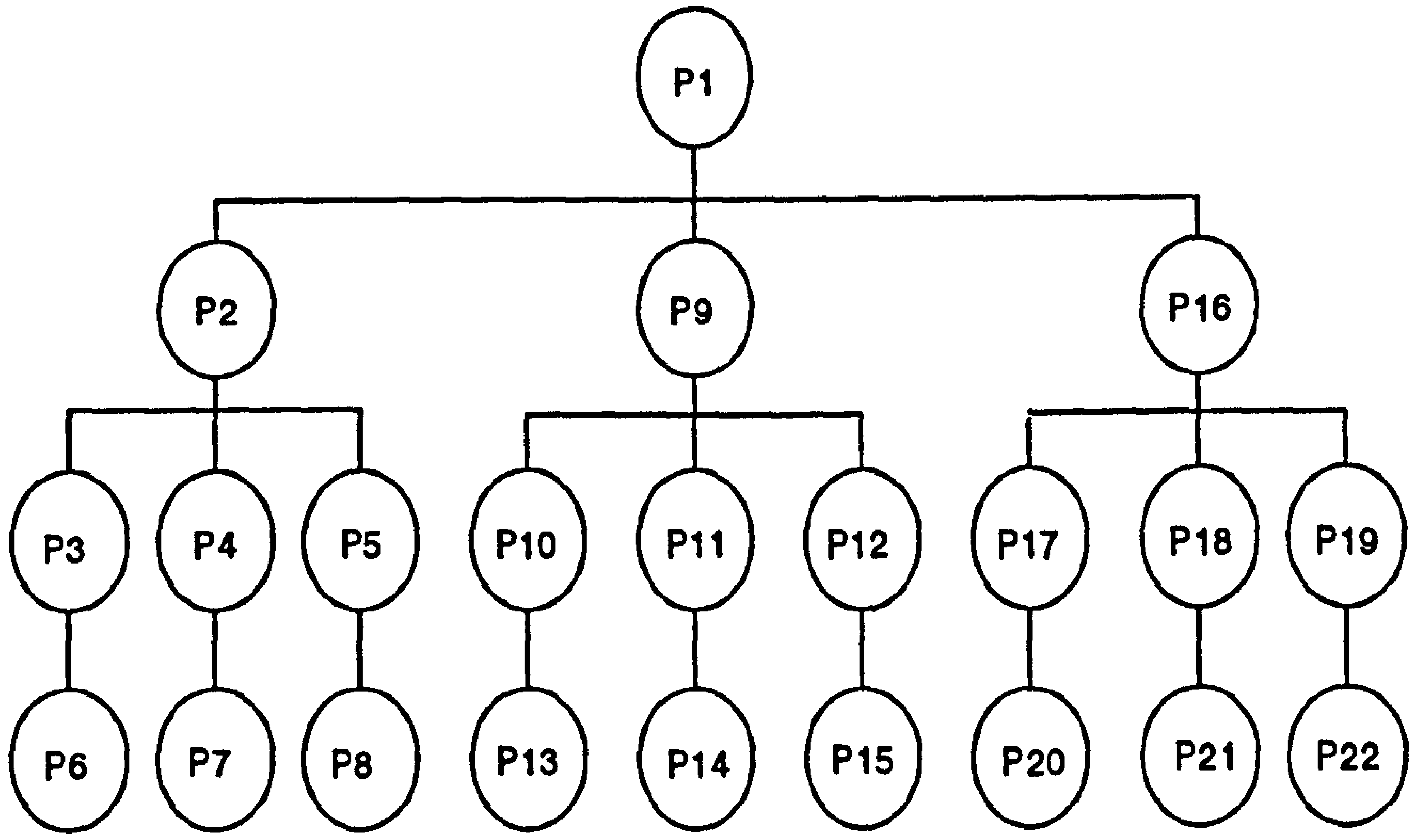


MIXED STRUCTURE

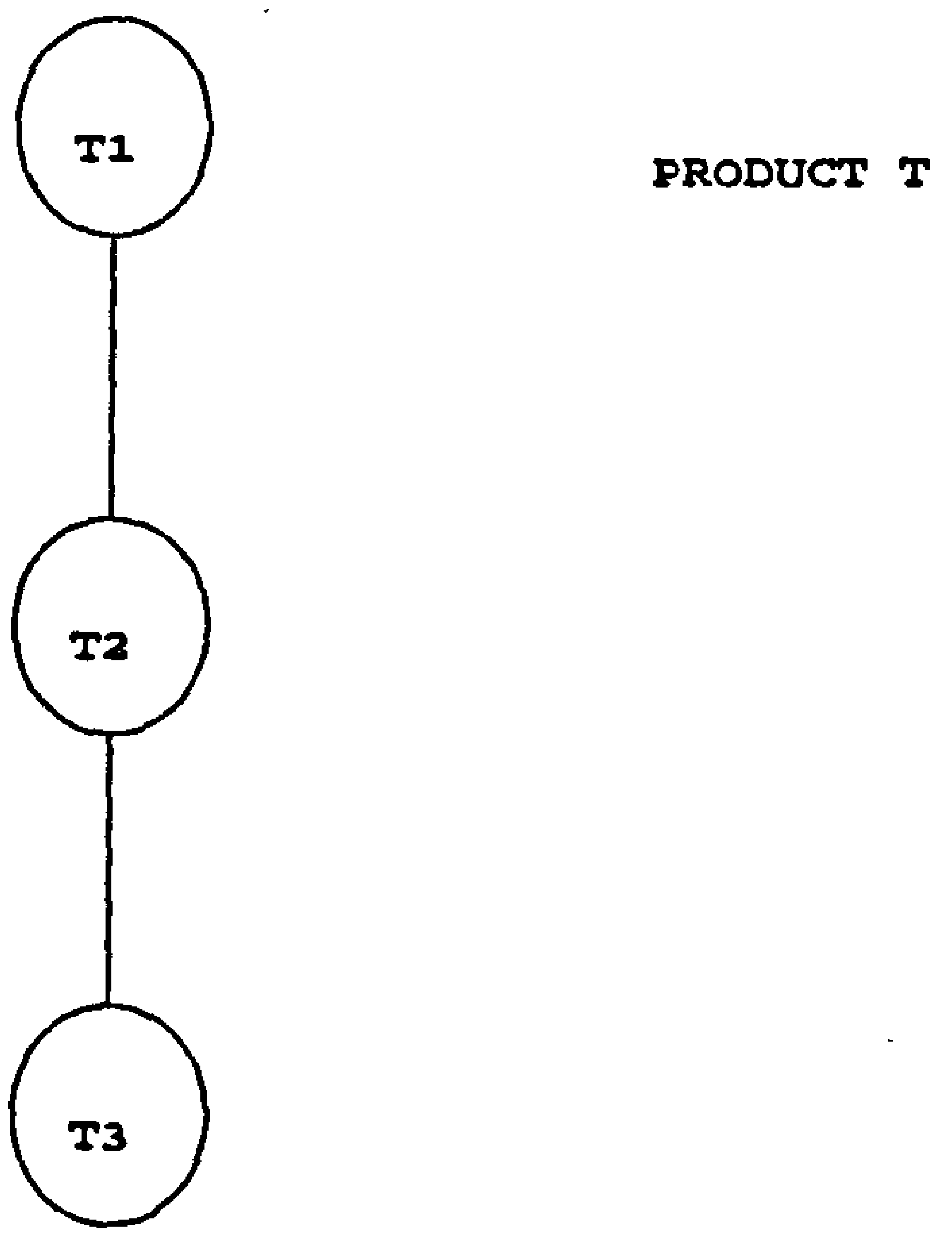


MIXED STRUCTURE

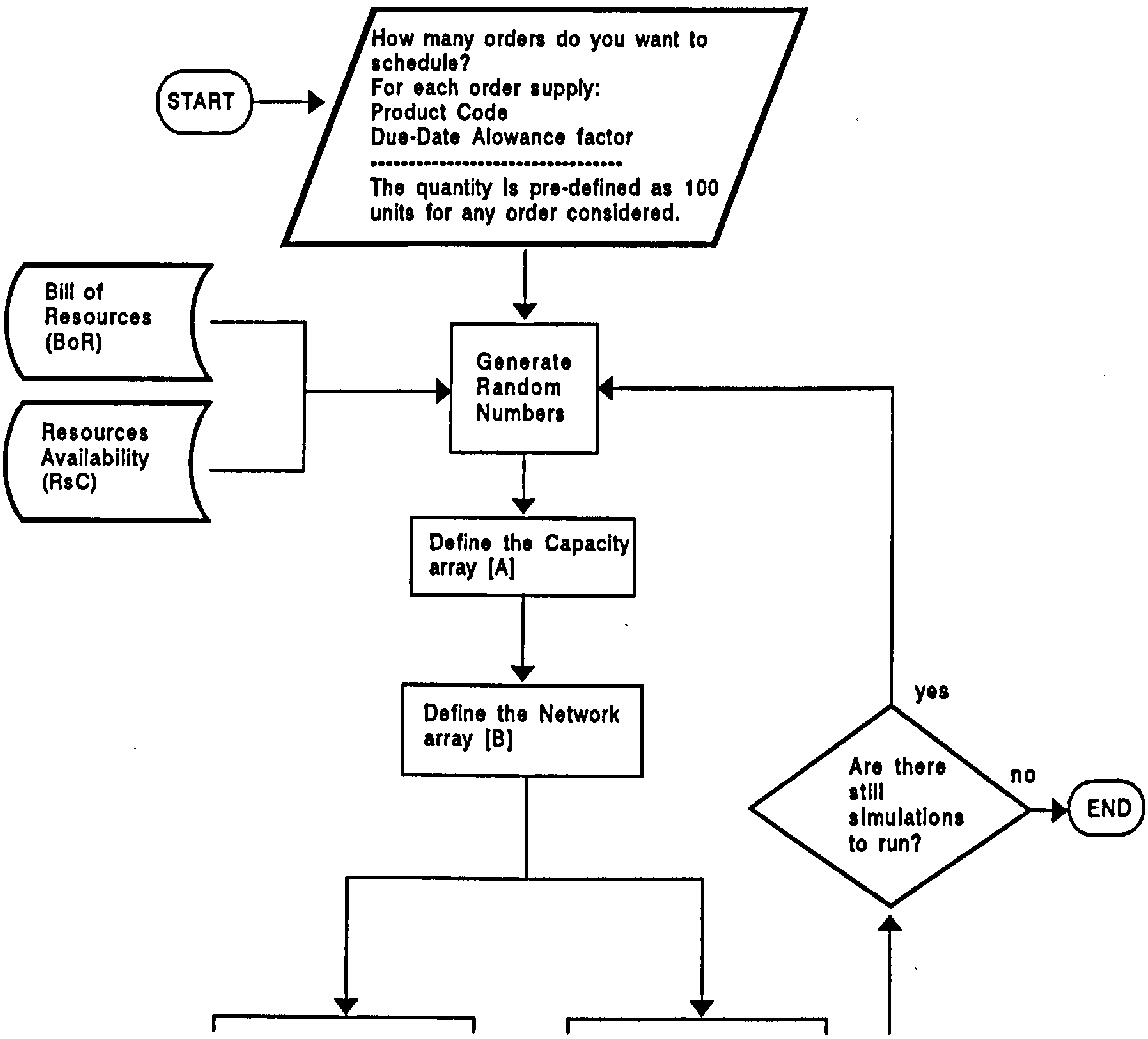
PRODUCT P



STRING TYPE JOB STRUCTURE



THE SIMULATION MODEL



APPENDIX 6: SIMULATION RESULTS

100 observation per replications

PRODUCT A Due-Date Factor = 1.2

r	RULE	MEAN FLOW TIME	MEAN WAITING TIME	MEAN TARDINESS	MEAN ABSOLUTE LATENESS	% of TARDY JOBS
1	TWK-FB	180.53	14.78	4.81	26.31	23.00
	IR-TWK	183.98	42.68	7.21	27.66	28.00
2	TWK-FB	190.99	18.84	11.16	31.41	29.00
	IR-TWK	192.75	50.75	11.01	29.35	39.00
3	TWK-FB	187.71	21.17	9.69	29.26	30.00
	IR-TWK	188.69	46.14	9.28	27.47	35.00
4	TWK-FB	181.26	16.20	5.12	25.67	23.00
	IR-TWK	186.53	43.53	8.59	27.33	32.00
5	TWK-FB	188.74	17.91	8.77	28.25	25.00
	IR-TWK	190.68	45.37	9.62	28.01	32.00
6	TWK-FB	187.95	20.54	10.24	29.10	27.00
	IR-TWK	189.15	45.01	10.08	27.58	30.00
7	TWK-FB	189.53	19.24	10.24	28.43	29.00
	IR-TWK	191.48	44.09	11.09	28.17	38.00
8	TWK-FB	185.57	15.47	7.45	27.46	26.00
	IR-TWK	188.20	43.56	9.48	28.88	31.00
9	TWK-FB	186.36	16.97	9.85	29.78	25.00
	IR-TWK	186.18	44.54	8.55	27.36	30.00
10	TWK-FB	185.23	15.79	7.92	31.17	18.00
	IR-TWK	189.44	42.83	9.39	29.89	28.00

r: REPLICATION

PRODUCT B - Due-date Factor = 1.2

r	RULE	MEAN FLOW TIME	MEAN WAITING TIME	MEAN TARDINESS	MEAN ABSOLUTE LATENESS	% of TARDY JOBS
1	TWK-FB	248.85	40.73	18.96	34.56	42.00
	IR-TWK	253.42	83.07	21.16	34.38	57.00
2	TWK-FB	253.67	46.36	24.80	37.44	55.00
	IR-TWK	262.19	91.69	28.96	37.25	71.00
3	TWK-FB	249.25	44.39	19.42	37.53	38.00
	IR-TWK	257.76	89.16	22.58	35.35	54.00
4	TWK-FB	254.89	44.19	25.09	40.02	45.00
	IR-TWK	261.74	90.40	29.25	41.48	57.00
5	TWK-FB	250.68	44.07	18.98	35.54	42.00
	IR-TWK	262.83	89.08	23.17	31.75	66.00
6	TWK-FB	248.51	41.03	18.76	35.73	40.00
	IR-TWK	260.25	88.80	26.72	39.91	60.00
7	TWK-FB	246.93	43.81	21.59	35.87	45.00
	IR-TWK	250.92	83.37	22.23	33.16	55.00
8	TWK-FB	261.06	48.73	27.82	41.60	47.00
	IR-TWK	270.46	97.83	34.57	45.70	61.00
9	TWK-FB	252.79	44.89	21.69	37.83	46.00
	IR-TWK	258.67	87.06	25.34	39.26	57.00
10	TWK-FB	255.89	53.29	27.49	39.49	54.00
	IR-TWK	253.73	86.30	25.81	38.28	53.00

PRODUCT C - Due-Date Factor = 1.2

r	RULE	MEAN FLOW TIME	MEAN WAITING TIME	MEAN TARDINESS	MEAN ABSOLUTE LATENESS	% of TARDY JOBS
1	TWK-FB	268.25	13.92	1.61	41.60	10.00
	IR-TWK	265.05	30.80	0.63	42.84	6.00
2	TWK-FB	269.09	15.24	4.84	44.42	11.00
	IR-TWK	268.46	40.10	3.40	42.16	10.00
3	TWK-FB	268.17	14.15	3.96	42.57	11.00
	IR-TWK	266.83	31.19	2.45	40.89	11.00
4	TWK-FB	267.69	11.49	3.13	42.34	11.00
	IR-TWK	267.21	31.92	3.04	42.66	12.00
5	TWK-FB	275.18	15.05	5.84	44.39	13.00
	IR-TWK	270.97	30.17	2.83	42.58	10.00
6	TWK-FB	267.02	13.71	3.10	41.24	10.00
	IR-TWK	265.03	28.71	1.36	39.74	7.00
7	TWK-FB	265.65	15.63	1.72	41.00	12.00
	IR-TWK	266.44	37.56	2.77	42.32	16.00
8	TWK-FB	273.21	13.95	2.99	41.40	8.00
	IR-TWK	273.28	31.45	2.50	40.35	12.00
9	TWK-FB	265.60	14.59	1.89	43.53	8.00
	IR-TWK	267.64	38.55	2.24	42.18	12.00
10	TWK-FB	267.10	11.40	1.99	38.56	12.00
	IR-TWK	265.17	24.04	1.60	39.71	10.00

PRODUCT D - Due_date Factor = 1.2

r	RULE	MEAN FLOW TIME	MEAN WAITING TIME	MEAN TARDINESS	MEAN ABSOLUTE LATENESS	% of TARDY JOBS
1	TWK-FB	325.28	52.74	24.37	42.78	44.00
	IR-TWK	328.63	69.50	23.35	37.40	56.00
2	TWK-FB	331.55	54.04	26.84	43.48	52.00
	IR-TWK	320.20	66.54	18.50	38.15	46.00
3	TWK-FB	341.78	56.22	29.89	43.55	54.00
	IR-TWK	332.39	69.07	23.26	39.69	53.00
4	TWK-FB	344.83	57.75	25.60	40.41	50.00
	IR-TWK	338.39	77.45	21.18	38.00	50.00
5	TWK-FB	337.27	57.89	31.56	47.78	50.00
	IR-TWK	322.00	63.52	18.45	36.81	49.00
6	TWK-FB	333.95	49.80	28.43	41.63	56.00
	IR-TWK	331.46	70.48	24.02	35.31	64.00
7	TWK-FB	333.91	48.43	25.70	40.44	51.00
	IR-TWK	328.98	66.96	20.55	35.09	58.00
8	TWK-FB	337.90	55.46	33.87	48.45	55.00
	IR-TWK	332.20	70.52	29.90	46.22	54.00
9	TWK-FB	332.09	53.96	23.47	39.81	47.00
	IR-TWK	322.49	69.75	15.07	32.60	46.00
10	TWK-FB	335.25	53.46	31.05	47.75	44.00
	IR-TWK	329.65	74.55	24.44	40.12	52.00

PRODUCT G : Due-Date Factor = 1.5

r	RULE	MEAN FLOW TIME	MEAN WAITING TIME	MEAN TARDINESS	MEAN ABSOLUTE LATENESS	% of TARDY JOBS
1	TWK-FB	449.25	153.99	88.03	92.81	88.00
	IR-TWK	422.87	199.11	62.81	68.74	82.00
2	TWK-FB	473.46	169.83	111.38	113.45	92.00
	IR-TWK	444.68	219.34	83.87	87.22	88.00
3	TWK-FB	458.56	165.40	104.17	107.04	92.00
	IR-TWK	440.50	213.61	86.97	90.69	88.00
4	TWK-FB	454.03	161.42	91.46	95.16	89.00
	IR-TWK	420.65	194.46	59.75	65.13	77.00
5	TWK-FB	437.99	149.99	74.29	81.61	80.00
	IR-TWK	420.19	193.48	57.30	65.42	77.00
6	TWK-FB	451.22	163.70	92.50	94.83	92.00
	IR-TWK	430.64	206.32	73.80	78.01	84.00
7	TWK-FB	450.24	163.52	95.78	99.25	90.00
	IR-TWK	424.53	202.03	71.89	77.20	84.00
8	TWK-FB	458.56	159.98	105.46	109.20	88.00
	IR-TWK	436.29	210.43	85.21	90.97	84.00
9	TWK-FB	460.94	163.58	97.20	98.98	91.00
	IR-TWK	436.91	209.89	76.65	81.91	85.00
10	TWK-FB	460.79	158.64	96.87	99.40	93.00
	IR-TWK	437.74	211.24	75.36	79.41	88.00

PRODUCT H: Due-Date Factor = 1.2

r	RULE	MEAN FLOW TIME	MEAN WAITING TIME	MEAN TARDINESS	MEAN ABSOLUTE LATENESS	% of TARDY JOBS
1	TWK-FB	546.78	148.73	50.72	57.52	76.00
	IR-TWK	546.94	139.10	53.08	62.07	71.00
2	TWK-FB	557.76	138.01	57.01	65.95	69.00
	IR-TWK	551.85	147.81	53.55	64.95	71.00
3	TWK-FB	558.88	149.89	60.99	68.99	70.00
	IR-TWK	551.95	141.63	55.05	64.03	71.00
4	TWK-FB	546.59	136.24	49.31	61.42	61.00
	IR-TWK	546.71	137.96	48.14	58.96	65.00
5	TWK-FB	547.39	145.79	52.77	62.87	69.00
	IR-TWK	541.89	134.43	49.26	61.37	64.00
6	TWK-FB	557.28	150.09	57.58	65.90	74.00
	IR-TWK	543.66	138.57	45.98	56.31	69.00
7	TWK-FB	545.93	143.13	55.87	64.27	70.00
	IR-TWK	539.14	136.28	52.34	64.00	65.00
8	TWK-FB	558.96	147.80	62.77	73.69	76.00
	IR-TWK	544.35	141.85	48.32	59.40	66.00
9	TWK-FB	551.43	139.82	49.86	59.42	70.00
	IR-TWK	545.72	139.21	46.04	57.50	67.00
10	TWK-FB	561.64	148.21	66.91	77.54	70.00
	IR-TWK	551.42	143.74	57.18	68.29	70.00

PRODUCT J : Due-Date Factor = 1.2

r	RULE	MEAN FLOW TIME	MEAN WAITING TIME	MEAN TARDINESS	MEAN ABSOLUTE LATENESS	% of TARDY JOBS
1	TWK-FB	302.89	41.04	11.66	41.06	26.00
	IR-TWK	300.95	67.58	8.51	36.69	30.00
2	TWK-FB	301.79	43.38	11.07	39.22	28.00
	IR-TWK	301.74	69.11	8.96	35.05	32.00
3	TWK-FB	307.08	46.37	12.01	41.14	30.00
	IR-TWK	302.37	70.31	7.73	37.29	30.00
4	TWK-FB	317.51	44.36	16.32	45.64	29.00
	IR-TWK	313.21	71.89	9.87	37.05	31.00
5	TWK-FB	303.16	44.40	11.54	39.98	29.00
	IR-TWK	298.08	60.28	7.52	37.03	27.00
6	TWK-FB	303.72	45.61	12.50	40.20	25.00
	IR-TWK	301.09	63.88	5.65	29.12	26.00
7	TWK-FB	307.43	48.21	10.81	39.55	23.00
	IR-TWK	305.96	70.74	8.07	35.55	33.00
8	TWK-FB	294.90	39.16	6.66	39.63	18.00
	IR-TWK	298.48	65.11	6.96	36.66	24.00
9	TWK-FB	308.79	44.23	14.13	44.27	27.00
	IR-TWK	301.88	67.79	6.72	36.37	30.00
10	TWK-FB	301.90	39.66	9.20	40.54	24.00
	IR-TWK	306.02	68.77	8.46	34.95	26.00

PRODUCT K Due-Date Factor = 1.2

r	RULE	MEAN FLOW TIME	MEAN WAITING TIME	MEAN TARDINESS	MEAN ABSOLUTE LATENESS	% of TARDY JOBS
1	TWK-FB	432.13	89.10	27.01	51.20	49.00
	IR-TWK	412.94	100.98	15.03	46.42	36.00
2	TWK-FB	423.96	90.99	24.40	45.57	50.00
	IR-TWK	412.51	102.60	16.93	42.06	37.00
3	TWK-FB	434.11	93.62	31.18	57.31	44.00
	IR-TWK	421.17	104.20	21.02	49.92	36.00
4	TWK-FB	420.75	90.02	22.68	46.29	42.00
	IR-TWK	411.62	101.04	16.50	43.06	42.00
5	TWK-FB	432.87	103.64	31.95	51.41	49.00
	IR-TWK	422.08	108.24	19.18	36.66	48.00
6	TWK-FB	441.14	97.47	35.72	52.20	52.00
	IR-TWK	432.79	116.95	29.50	48.12	48.00
7	TWK-FB	444.48	105.41	29.50	48.17	53.00
	IR-TWK	427.45	110.53	18.07	42.33	47.00
8	TWK-FB	438.15	103.26	31.43	49.76	52.00
	IR-TWK	426.73	110.25	21.29	40.91	50.00
9	TWK-FB	428.17	96.20	26.99	46.46	44.00
	IR-TWK	422.37	109.47	21.96	42.19	47.00
10	TWK-FB	431.76	98.44	28.99	48.39	51.00
	IR-TWK	420.80	112.12	20.44	42.24	46.00

PRODUCT L : Due-Date = 1.2

r	RULE	MEAN FLOW TIME	MEAN WAITING TIME	MEAN TARDINESS	MEAN ABSOLUTE LATENESS	% of TARDY JOBS
1	TWK-FB	243.84	14.18	6.08	34.02	17.00
	IR-TWK	239.12	21.74	2.98	32.55	21.00
2	TWK-FB	243.31	14.43	7.19	37.30	21.00
	IR-TWK	241.54	24.31	4.05	32.81	19.00
3	TWK-FB	245.81	17.07	7.65	33.91	17.00
	IR-TWK	241.56	24.89	3.90	30.65	21.00
4	TWK-FB	248.73	18.20	9.42	34.39	26.00
	IR-TWK	243.65	24.78	6.05	32.73	24.00
5	TWK-FB	248.93	16.07	9.84	36.17	24.00
	IR-TWK	247.62	25.79	7.24	32.30	30.00
6	TWK-FB	238.69	16.61	8.24	31.59	26.00
	IR-TWK	234.77	23.06	6.04	31.09	30.00
7	TWK-FB	245.66	18.16	10.37	34.55	26.00
	IR-TWK	239.00	23.77	4.90	30.27	24.00
8	TWK-FB	246.20	15.76	7.82	35.71	22.00
	IR-TWK	242.56	24.88	5.39	34.49	22.00
9	TWK-FB	238.52	15.13	7.76	35.60	20.00
	IR-TWK	236.64	22.84	5.08	32.16	22.00
10	TWK-FB	248.90	18.64	9.80	32.93	33.00
	IR-TWK	242.98	24.04	5.31	29.87	28.00

PRODUCT M Due-Date Factor = 1.5

r	RULE	MEAN FLOW TIME	MEAN WAITING TIME	MEAN TARDINESS	MEAN ABSOLUTE LATENESS	% of TARDY JOBS
1	TWK-FB	379.81	109.03	97.23	98.28	93.00
	IR-TWK	352.26	136.72	71.53	74.44	87.00
2	TWK-FB	382.08	115.41	97.22	98.37	93.00
	IR-TWK	360.42	139.41	76.05	77.70	93.00
3	TWK-FB	374.61	106.97	91.89	93.57	92.00
	IR-TWK	359.07	141.88	77.35	80.02	88.00
4	TWK-FB	374.11	107.66	95.66	96.31	93.00
	IR-TWK	359.68	143.63	81.21	81.84	94.00
5	TWK-FB	370.11	106.29	90.17	91.04	95.00
	IR-TWK	338.71	122.33	59.80	61.71	89.00
6	TWK-FB	388.07	112.79	105.46	105.91	96.00
	IR-TWK	357.76	135.98	75.15	75.60	96.00
7	TWK-FB	377.25	106.79	95.61	97.83	87.00
	IR-TWK	355.01	138.58	73.29	75.42	90.00
8	TWK-FB	368.56	109.30	86.71	87.73	94.00
	IR-TWK	349.61	131.04	69.25	71.78	88.00
9	TWK-FB	378.66	111.18	96.18	96.90	95.00
	IR-TWK	357.46	138.79	75.12	75.98	93.00
10	TWK-FB	371.17	107.11	92.51	94.73	93.00
	IR-TWK	355.69	138.95	77.33	79.84	91.00

PRODUCT A B : Due-Date Factor = 1.5

r	RULE	MEAN FLOW TIME	MEAN WAITING TIME	MEAN TARDINESS	MEAN ABSOLUTE LATENESS	% of TARDY JOBS
1	TWK-FB	268.37	58.74	22.73	54.78	37.00
	IR-TWK	269.94	96.19	25.17	58.11	38.50
2	TWK-FB	277.08	63.70	26.73	55.57	41.50
	IR-TWK	280.50	103.77	31.70	62.07	43.50
3	TWK-FB	279.94	69.91	30.09	56.29	44.50
	IR-TWK	280.88	107.57	33.38	61.93	47.50
4	TWK-FB	271.64	62.31	26.01	55.32	43.00
	IR-TWK	273.56	96.40	29.61	60.59	39.00
5	TWK-FB	278.34	65.89	29.24	58.67	41.00
	IR-TWK	275.95	98.95	29.76	62.09	38.50
6	TWK-FB	272.97	60.93	25.70	55.26	42.50
	IR-TWK	273.21	98.21	28.33	60.27	42.00
7	TWK-FB	273.66	66.08	27.32	55.76	40.50
	IR-TWK	272.62	99.33	28.57	59.32	42.00
8	TWK-FB	265.25	62.00	24.29	55.70	40.50
	IR-TWK	268.94	95.69	29.95	63.42	41.00
9	TWK-FB	277.38	61.43	27.41	57.83	39.00
	IR-TWK	277.76	103.31	28.38	59.39	37.50
10	TWK-FB	273.56	66.13	21.95	48.61	43.00
	IR-TWK	275.86	104.75	26.59	55.57	41.00

PRODUCT C D : Due-Date Factor = 1.5

r	RULE	MEAN FLOW TIME	MEAN WAITING TIME	MEAN TARDINESS	MEAN ABSOLUTE LATENESS	% of TARDY JOBS
1	TWK-FB	382.33	92.61	32.01	72.69	42.50
	IR-TWK	370.44	108.12	25.65	71.86	35.00
2	TWK-FB	385.92	95.12	36.60	78.40	42.00
	IR-TWK	369.57	104.58	25.84	73.23	37.50
3	TWK-FB	389.11	85.72	37.81	76.64	44.00
	IR-TWK	380.73	107.46	34.69	78.79	43.00
4	TWK-FB	386.59	88.63	36.74	78.79	43.50
	IR-TWK	371.37	104.61	31.07	82.67	41.00
5	TWK-FB	381.94	84.17	30.58	72.85	38.00
	IR-TWK	373.08	104.23	27.04	74.63	34.50
6	TWK-FB	391.08	86.67	39.57	84.90	43.00
	IR-TWK	380.68	111.23	32.87	81.80	37.00
7	TWK-FB	386.46	92.90	36.16	76.27	43.00
	IR-TWK	375.77	110.57	31.53	77.70	40.00
8	TWK-FB	387.87	93.36	38.23	80.94	47.50
	IR-TWK	381.53	115.54	33.22	77.26	43.00
9	TWK-FB	383.74	85.05	32.46	75.85	44.00
	IR-TWK	378.88	111.36	31.30	78.37	40.50
10	TWK-FB	381.91	89.37	33.07	75.73	37.00
	IR-TWK	374.55	108.59	30.26	77.48	38.00

PRODUCT L G M Due-Date Factor = 1.5

r	RULE	MEAN FLOW TIME	MEAN WAITING TIME	MEAN TARDINESS	MEAN ABSOLUTE LATENESS	% of TARDY JOBS
1	TWK-FB	587.01	272.81	241.30	247.40	87.67
	IR-TWK	531.98	307.84	200.80	220.79	70.00
2	TWK-FB	583.05	266.33	239.43	244.22	90.67
	IR-TWK	517.71	296.31	190.44	211.57	70.67
3	TWK-FB	581.75	229.25	237.69	241.56	89.67
	IR-TWK	523.09	171.18	195.32	215.47	70.67
4	TWK-FB	589.28	279.82	247.17	251.74	86.00
	IR-TWK	524.77	205.39	198.97	219.86	70.33
5	TWK-FB	579.50	269.39	236.15	239.41	90.33
	IR-TWK	517.37	191.71	190.90	211.04	70.00
6	TWK-FB	582.27	234.92	241.36	246.94	88.33
	IR-TWK	530.90	172.26	203.01	221.61	71.33
7	TWK-FB	571.46	225.59	229.35	234.43	85.67
	IR-TWK	519.00	171.02	193.63	215.45	69.33
8	TWK-FB	595.72	270.14	254.78	260.68	87.33
	IR-TWK	516.54	197.99	191.85	214.00	70.00
9	TWK-FB	575.27	272.93	234.40	241.22	85.67
	IR-TWK	523.55	203.94	197.79	219.54	68.67
10	TWK-FB	578.26	227.98	239.97	246.02	87.67
	IR-TWK	521.18	167.25	197.53	218.23	71.67

PRODUCT J K H Due-Date Factor = 1.5

r	RULE	MEAN FLOW TIME	MEAN WAITING TIME	MEAN TARDINESS	MEAN ABSOLUTE LATENESS	% of TARDY JOBS
1	TWK-FB	686.76	304.03	166.64	174.11	83.00
	IR-TWK	642.68	235.52	139.31	163.53	67.33
2	TWK-FB	680.97	298.96	167.55	175.46	83.33
	IR-TWK	627.38	230.78	133.42	160.78	64.00
3	TWK-FB	693.39	321.25	175.73	183.48	84.00
	IR-TWK	649.47	240.39	148.53	172.99	67.33
4	TWK-FB	687.36	304.41	167.49	174.96	83.00
	IR-TWK	643.17	233.63	139.79	163.76	67.33
5	TWK-FB	691.47	306.04	172.06	179.52	83.00
	IR-TWK	646.46	237.99	143.50	167.42	67.33
6	TWK-FB	690.65	299.60	177.32	185.92	84.00
	IR-TWK	648.21	241.98	147.88	169.50	70.67
7	TWK-FB	668.20	292.17	159.23	169.82	78.67
	IR-TWK	633.94	234.49	140.09	165.78	65.33
8	TWK-FB	671.27	293.08	158.72	167.27	82.00
	IR-TWK	639.85	242.15	143.39	168.03	66.33
9	TWK-FB	674.53	292.20	165.38	173.67	81.67
	IR-TWK	634.60	232.68	140.94	164.74	67.00
10	TWK-FB	674.12	294.42	160.47	172.07	79.00
	IR-TWK	642.01	234.76	142.45	168.16	64.00

PRODUCT A L B M G Due-Date Factor = 2.5

r	RULE	MEAN FLOW TIME	MEAN WAITING TIME	MEAN TARDINESS	MEAN ABSOLUTE LATENESS	% of TARDY JOBS
1	TWK-FB	617.81	293.82	134.42	186.66	55.40
	IR-TWK	574.62	326.31	120.33	201.68	51.20
2	TWK-FB	637.79	301.89	146.78	192.20	61.80
	IR-TWK	560.95	305.61	111.34	198.14	45.40
3	TWK-FB	627.26	307.98	147.36	194.21	61.80
	IR-TWK	563.08	315.36	116.16	195.99	47.80
4	TWK-FB	618.36	299.17	137.08	190.30	53.40
	IR-TWK	575.37	327.79	120.90	200.92	48.00
5	TWK-FB	632.25	300.38	146.59	192.43	59.80
	IR-TWK	569.44	318.10	119.67	201.40	46.20
6	TWK-FB	626.35	296.55	144.63	195.74	59.60
	IR-TWK	561.80	307.86	112.57	196.17	49.60
7	TWK-FB	620.85	298.32	138.34	187.71	58.80
	IR-TWK	558.57	307.66	110.55	194.40	49.00
8	TWK-FB	610.58	286.67	132.63	185.36	58.00
	IR-TWK	557.50	306.79	110.62	194.42	47.60
9	TWK-FB	632.13	307.02	146.53	195.21	59.60
	IR-TWK	558.18	304.36	109.30	194.68	45.40
10	TWK-FB	618.40	303.92	136.73	183.84	59.80
	IR-TWK	558.66	318.28	110.46	191.05	47.60

PRODUCT C J D K H Due-Date Factor: 2.5

r	RULE	MEAN FLOW TIME	MEAN WAITING TIME	MEAN TARDINESS	MEAN ABSOLUTE LATENESS	% of TARDY JOBS
1	TWK-FB	776.52	362.09	76.41	161.49	45.60
	IR-TWK	721.23	275.33	63.35	190.67	36.80
2	TWK-FB	772.85	369.80	72.66	151.99	48.00
	IR-TWK	704.56	271.22	50.75	176.46	34.40
3	TWK-FB	785.14	377.93	81.73	156.20	54.00
	IR-TWK	711.97	273.74	60.36	186.64	37.80
4	TWK-FB	782.22	372.34	79.28	158.77	48.20
	IR-TWK	717.15	284.04	58.42	182.14	34.60
5	TWK-FB	750.35	352.90	62.02	153.04	40.80
	IR-TWK	705.34	276.82	54.27	182.55	32.60
6	TWK-FB	768.75	362.35	74.25	155.27	47.80
	IR-TWK	708.08	268.94	56.62	180.68	35.80
7	TWK-FB	769.57	357.58	67.52	152.58	46.20
	IR-TWK	722.86	293.78	61.16	186.57	35.80
8	TWK-FB	773.36	363.11	69.27	152.75	45.40
	IR-TWK	716.62	270.92	54.35	179.65	35.80
9	TWK-FB	778.50	369.58	76.44	154.11	48.00
	IR-TWK	715.54	279.18	58.70	181.58	38.20
10	TWK-FB	760.01	357.65	66.74	154.31	44.60
	IR-TWK	695.18	265.89	43.95	173.56	30.60

PRODUCT E P F Due-Date Factor = 2.0

r	RULE	MEAN FLOW TIME	MEAN WAITING TIME	MEAN TARDINESS	MEAN ABSOLUTE LATENESS	% of TARDY JOBS
1	TWK-FB	957.20	374.50	271.13	289.39	82.33
	IR-TWK	831.57	404.09	172.15	217.06	69.67
2	TWK-FB	956.63	374.87	274.60	289.17	86.67
	IR-TWK	811.38	392.34	160.65	206.51	69.00
3	TWK-FB	961.30	373.37	275.28	290.04	85.00
	IR-TWK	823.36	397.97	165.88	209.20	71.33
4	TWK-FB	931.55	354.89	253.72	272.13	82.00
	IR-TWK	818.72	396.95	169.11	215.75	68.33
5	TWK-FB	947.34	362.86	264.89	282.77	83.00
	IR-TWK	819.06	389.61	166.14	213.56	66.33
6	TWK-FB	979.43	387.81	295.28	312.07	85.33
	IR-TWK	827.85	400.58	169.46	212.00	70.33
7	TWK-FB	990.72	392.61	306.33	320.03	85.67
	IR-TWK	819.72	396.49	168.45	215.29	68.67
8	TWK-FB	942.71	367.92	262.70	283.54	79.33
	IR-TWK	824.76	398.47	172.38	220.85	67.67
9	TWK-FB	974.89	384.00	292.35	306.06	84.67
	IR-TWK	817.92	395.44	168.10	214.54	68.67
10	TWK-FB	931.96	360.37	257.39	273.73	82.00
	IR-TWK	824.06	400.67	174.43	215.71	68.67

PRODUCTS A B C D E F G H J K L M N P Due-Date Factor: 2.5

r	RULE	MEAN FLOW TIME	MEAN WAITING TIME	MEAN TARDINESS	MEAN ABSOLUTE LATENESS	% of TARDY JOBS
1	TWK-FB	1652.80	854.41	591.64	632.24	78.57
	IR-TWK	1374.21	728.82	395.81	518.78	57.14
2	TWK-FB	1596.53	803.82	399.47	454.89	71.43
	IR-TWK	1353.72	701.21	406.78	592.31	57.14
3	TWK-FB	1574.88	782.12	453.28	488.31	78.57
	IR-TWK	1475.07	709.66	565.76	713.07	64.29
4	TWK-FB	1646.66	825.00	621.35	772.04	64.29
	IR-TWK	1405.53	726.31	432.11	564.67	57.14
5	TWK-FB	1601.71	795.90	506.82	620.06	54.29
	IR-TWK	1483.06	745.41	449.79	624.67	50.29
6	TWK-FB	1620.12	844.59	596.22	729.78	60.00
	IR-TWK	1381.66	781.42	576.64	729.09	52.00
7	TWK-FB	1685.84	826.79	675.25	763.85	64.29
	IR-TWK	1436.41	720.95	589.25	731.28	58.20
8	TWK-FB	1852.02	900.21	720.46	782.25	85.71
	IR-TWK	1482.71	780.60	436.98	584.60	71.43
9	TWK-FB	1653.18	897.18	602.05	662.91	66.39
	IR-TWK	1491.55	712.57	548.08	633.66	60.21
10	TWK-FB	1674.54	843.84	624.18	646.77	78.57
	IR-TWK	1377.07	766.56	445.58	601.24	50.00

APPENDIX 7 - STATISTICAL TESTS

GROUP	MEASURES	t_f^1	f	t_f	DECISION
A	MEAN FLOW TIME	1.7099	14.9529	1.753	accepted
	MEAN WAITING TIME	26.6807	15.9223	1.746	rejected
	MEAN ABSOLUTE LATENESS	0.7733	10.9429	1.797	accepted
	MEAN TARDINESS	1.1649	11.7321	1.786	accepted
	MEAN PERCENTAGE OF TARDY JOBS	4.0740	15.9132	1.747	rejected
B	MEAN FLOW TIME	3.0921	14.7471	1.755	rejected
	MEAN WAITING TIME	24.4482	15.6025	1.749	rejected
	MEAN ABSOLUTE LATENESS	0.0598	11.7675	1.785	accepted
	MEAN TARDINESS	2.0484	15.6692	1.748	rejected
	MEAN PERCENTAGE OF TARDY JOBS	5.4881	15.9904	1.746	rejected
C	MEAN FLOW TIME	0.8339	15.6207	1.749	accepted
	MEAN WAITING TIME	11.4581	8.5369	1.846	rejected
	MEAN ABSOLUTE LATENESS	0.8195	14.1313	1.755	accepted
	MEAN TARDINESS	1.5881	12.7178	1.774	accepted
	MEAN PERCENTAGE OF TARDY JOBS	0.0000	12.5661	1.776	accepted
D	MEAN FLOW TIME	2.7105	15.9916	1.746	rejected
	MEAN WAITING TIME	9.9821	15.0375	1.753	rejected
	MEAN ABSOLUTE LATENESS	3.6434	15.7637	1.748	rejected
	MEAN TARDINESS	3.6709	15.4629	1.750	rejected
	MEAN PERCENTAGE OF TARDY JOBS	1.1285	14.8101	1.754	accepted
G	MEAN FLOW TIME	5.8632	15.9547	1.746	rejected
	MEAN WAITING TIME	13.8751	13.7537	1.763	rejected
	MEAN ABSOLUTE LATENESS	5.7103	15.9433	1.746	rejected
	MEAN TARDINESS	2.4551	9.3451	1.826	rejected
	MEAN PERCENTAGE OF TARDY JOBS	0.0568	15.8919	1.747	accepted
H	MEAN FLOW TIME	2.8767	14.1243	1.760	rejected
	MEAN WAITING TIME	2.3128	14.7427	1.755	rejected
	MEAN ABSOLUTE LATENESS	1.7732	12.6873	1.774	accepted
	MEAN TARDINESS	2.4609	13.5602	1.765	rejected
	MEAN PERCENTAGE OF TARDY JOBS	1.5941	13.4353	1.767	accepted
J	MEAN FLOW TIME	0.8317	14.7318	1.755	accepted
	MEAN WAITING TIME	16.4996	15.3503	1.750	rejected
	MEAN ABSOLUTE LATENESS	5.4278	15.7341	1.748	rejected
	MEAN TARDINESS	4.1261	10.7346	1.800	rejected
	MEAN PERCENTAGE OF TARDY JOBS	2.0339	15.3470	1.750	rejected
K	MEAN FLOW TIME	3.6530	15.9642	1.746	rejected
	MEAN WAITING TIME	4.3134	15.7694	1.748	rejected
	MEAN ABSOLUTE LATENESS	3.5892	15.8370	1.747	rejected
	MEAN TARDINESS	5.0818	15.9564	1.746	rejected
	MEAN PERCENTAGE OF TARDY JOBS	2.3057	14.2432	1.759	rejected

L	MEAN FLOW TIME	2.3246	15.9752	1.746	rejected
	MEAN WAITING TIME	12.0490	14.7412	1.755	rejected
	MEAN ABSOLUTE LATENESS	3.9705	15.5500	1.749	rejected
	MEAN TARDINESS	5.7005	15.7603	1.746	rejected
	MEAN PERCENTAGE OF TARDY JOBS	0.4548	15.2251	1.751	accepted
M	MEAN FLOW TIME	7.8052	15.8771	1.747	rejected
	MEAN WAITING TIME	12.7835	11.1130	1.794	rejected
	MEAN ABSOLUTE LATENESS	8.7275	15.5995	1.749	rejected
	MEAN TARDINESS	8.6787	15.6026	1.749	rejected
	MEAN PERCENTAGE OF TARDY JOBS	1.7910	15.3643	1.750	rejected
AB	MEAN FLOW TIME	0.5683	15.7252	1.748	accepted
	MEAN WAITING TIME	21.9527	15.1264	1.752	rejected
	MEAN ABSOLUTE LATENESS	4.3848	15.6224	1.747	rejected
	MEAN TARDINESS	2.6921	15.7994	1.747	rejected
	MEAN PERCENTAGE OF TARDY JOBS	0.1719	14.5093	1.756	accepted
CD	MEAN FLOW TIME	5.7338	14.0903	1.760	rejected
	MEAN WAITING TIME	11.3758	15.8953	1.747	rejected
	MEAN ABSOLUTE LATENESS	0.0566	15.9143	1.746	accepted
	MEAN TARDINESS	3.5804	15.9794	1.746	rejected
	MEAN PERCENTAGE OF TARDY JOBS	2.5893	15.9993	1.746	rejected
LGM	MEAN FLOW TIME	21.2897	14.9306	1.753	rejected
	MEAN WAITING TIME	2.6190	10.2713	1.808	rejected
	MEAN ABSOLUTE LATENESS	11.0496	11.5959	1.788	rejected
	MEAN TARDINESS	17.0389	12.9808	1.771	rejected
	MEAN PERCENTAGE OF TARDY JOBS	27.2326	10.9146	1.797	rejected
JKH	MEAN FLOW TIME	10.7746	14.3586	1.758	rejected
	MEAN WAITING TIME	20.7009	10.4185	1.805	rejected
	MEAN ABSOLUTE LATENESS	4.2640	12.7259	1.774	rejected
	MEAN TARDINESS	10.1178	13.7255	1.764	rejected
	MEAN PERCENTAGE OF TARDY JOBS	18.0038	15.9930	1.746	rejected
AB GL M	MEAN FLOW TIME	11.8334	15.2580	1.751	rejected
	MEAN WAITING TIME	4.1718	14.6790	1.755	rejected
	MEAN ABSOLUTE LATENESS	3.7174	15.4423	1.750	rejected
	MEAN TARDINESS	11.4910	15.1262	1.752	rejected
	MEAN PERCENTAGE OF TARDY JOBS	10.8860	14.0763	1.760	rejected
CJ DH K	MEAN FLOW TIME	14.0475	15.3985	1.750	rejected
	MEAN WAITING TIME	24.9260	15.9498	1.746	rejected
	MEAN ABSOLUTE LATENESS	14.5444	12.7951	1.773	rejected
	MEAN TARDINESS	6.1924	15.8809	1.747	rejected
	MEAN PERCENTAGE OF TARDY JOBS	8.9850	13.9547	1.761	rejected
EFP	MEAN FLOW TIME	20.7428	8.4727	1.847	rejected
	MEAN WAITING TIME	5.9035	9.1210	1.830	rejected
	MEAN ABSOLUTE LATENESS	14.9565	8.1549	1.856	rejected
	MEAN TARDINESS	18.9164	7.9142	1.863	rejected
	MEAN PERCENTAGE OF TARDY JOBS	17.7104	12.9756	1.771	rejected

ABC	MEAN FLOW TIME	7.6922	14.0546	1.760	rejected
DEF	MEAN WAITING TIME	6.8327	15.8519	1.747	rejected
GHJ	MEAN ABSOLUTE LATENESS	0.3069	13.3160	1.768	accepted
KLM	MEAN TARDINESS	2.3965	14.8928	1.753	rejected
NP	MEAN PERCENTAGE OF TARDY JOBS	2.6519	13.5521	1.765	rejected

APPENDIX 8: SIMULATION RESULTS - OVERLAPPING MODE

PRODUCTS A and B
Due-Date Factor = 1.5 Overlapping level = 90%

r	RULE	MEAN FLOW TIME	MEAN TARDINESS	MEAN ABSOLUTE LATENESS	% of TARDY JOBS
1	TWK-FB	225.07	10.03	72.68	20.50
	IR-TWK	269.94	25.17	58.11	38.50
2	TWK-FB	234.73	14.66	73.77	26.50
	IR-TWK	280.50	31.70	62.07	43.50
3	TWK-FB	233.22	16.72	76.25	25.50
	IR-TWK	280.88	33.38	61.93	47.50
4	TWK-FB	228.18	12.28	71.32	20.50
	IR-TWK	273.56	29.61	60.59	39.00
5	TWK-FB	231.23	15.41	78.12	24.50
	IR-TWK	275.95	29.76	62.09	38.50
6	TWK-FB	228.53	12.80	73.89	24.00
	IR-TWK	273.21	28.33	60.27	42.00
7	TWK-FB	232.08	14.36	71.43	25.50
	IR-TWK	272.62	28.57	59.32	42.00
8	TWK-FB	223.69	13.67	76.10	26.00
	IR-TWK	268.94	29.95	63.42	41.00
9	TWK-FB	237.84	18.83	80.20	24.00
	IR-TWK	277.76	28.38	59.39	37.50
10	TWK-FB	232.60	11.19	68.03	21.00
	IR-TWK	275.86	26.59	55.57	41.00

FINAL ANSWER

	MEAN FLOW TIME	STANDARD DEVIATION FLOW TIME	MEAN TARDINESS	MEAN ABSOLUTE LATENESS	% of TARDY JOBS
TWK-FB	230.72	4.36	13.99	74.18	23.80
IR-TWK	274.92	4.04	29.14	60.28	41.05

PRODUCTS C and D
Due_date factor: 1.5 Overlapping Level % 0.90

r	RULE	MEAN FLOW TIME	MEAN TARDINESS	MEAN ABSOLUTE LATENESS	% of TARDY JOBS
1	TWK-FB	298.43	11.33	115.22	17.50
	IR-TWK	370.44	25.65	71.86	35.00
2	TWK-FB	295.16	12.02	120.00	19.50
	IR-TWK	369.57	25.84	73.23	37.50
3	TWK-FB	299.94	13.18	116.55	16.00
	IR-TWK	380.73	34.69	78.79	43.00
4	TWK-FB	301.82	9.85	111.53	18.00
	IR-TWK	373.08	27.04	74.63	34.50
5	TWK-FB	309.11	15.88	119.49	19.50
	IR-TWK	380.68	32.87	81.89	37.00
6	TWK-FB	296.79	11.09	115.79	17.00
	IR-TWK	375.77	31.53	77.70	40.00
7	TWK-FB	305.18	11.93	111.04	21.00
	IR-TWK	381.53	33.22	77.26	43.00
8	TWK-FB	307.41	11.66	110.57	19.50
	IR-TWK	378.88	31.30	78.37	40.50
9	TWK-FB	302.80	11.48	111.68	16.50
	IR-TWK	374.55	30.26	77.48	38.00
10	TWK-FB	300.22	12.19	116.05	18.00
	IR-TWK	371.37	31.07	82.67	41.00

FINAL ANSWER

	MEAN FLOW TIME	STANDARD DEVIATION FLOW TIME	MEAN TARDINESS	MEAN ABSOLUTE LATENESS	% of TARDY JOBS
TWK-FB	301.68	4.52	12.06	114.79	18.25
IR-TWK	375.66	4.55	30.35	77.39	38.95

PRODUCTS L, G and M
Due_date factor: 1.5 Overlapping Level % 0.90

r	RULE	MEAN FLOW TIME	MEAN TARDINESS	MEAN ABSOLUTE LATENESS	% of TARDY JOBS
1	TWK-FB	510.94	180.77	202.41	74.67
	IR-TWK	531.98	200.48	220.79	70.00
2	TWK-FB	488.53	163.91	187.70	71.67
	IR-TWK	517.71	190.44	211.57	70.67
3	TWK-FB	503.40	178.52	201.57	73.67
	IR-TWK	523.09	195.32	215.47	70.67
4	TWK-FB	503.43	179.10	201.45	73.00
	IR-TWK	524.77	198.97	219.86	70.33
5	TWK-FB	494.50	168.90	189.90	75.33
	IR-TWK	517.37	190.90	211.04	70.00
6	TWK-FB	504.33	179.19	200.55	75.67
	IR-TWK	530.90	203.01	221.61	71.33
7	TWK-FB	495.66	170.85	193.24	72.67
	IR-TWK	519.00	193.63	215.45	69.33
8	TWK-FB	503.29	177.23	198.00	75.33
	IR-TWK	516.54	191.85	214.00	70.00
9	TWK-FB	493.50	172.59	199.18	72.67
	IR-TWK	523.55	197.79	219.54	68.67
10	TWK-FB	503.66	179.13	198.94	75.33
	IR-TWK	521.18	197.53	218.23	71.67

FINAL ANSWER

		MEAN FLOW TIME	STANDARD DEVIATION FLOW TIME	MEAN TARDINESS	MEAN ABSOLUTE LATENESS	% of TARDY JOBS
1	TWK-FB	500.12	6.74	175.02	197.29	74.00
2	IR-TWK	522.61	5.43	195.99	216.76	70.27

PRODUCTS J, K and H
Due_date factor: 1.5 Overlapping Level % 0.90

r	RULE	MEAN FLOW TIME	MEAN TARDINESS	MEAN ABSOLUTE LATENESS	% of TARDY JOBS
1	TWK-FB	570.28	80.17	117.64	59.33
	IR-TWK	642.68	139.31	163.53	67.33
2	TWK-FB	549.55	73.24	118.25	54.00
	IR-TWK	627.38	133.42	160.78	64.00
3	TWK-FB	581.51	89.84	123.59	60.33
	IR-TWK	649.47	148.53	172.99	67.33
4	TWK-FB	554.63	76.39	115.56	53.67
	IR-TWK	631.65	136.86	159.50	66.67
5	TWK-FB	560.77	81.19	123.01	55.00
	IR-TWK	642.83	145.27	169.12	69.00
6	TWK-FB	572.05	87.97	125.81	55.67
	IR-TWK	643.73	145.45	169.08	68.67
7	TWK-FB	551.05	73.19	115.22	54.00
	IR-TWK	634.77	141.24	167.59	66.00
8	TWK-FB	555.04	75.55	118.46	52.33
	IR-TWK	630.06	133.28	158.88	67.33
9	TWK-FB	561.59	81.19	120.87	52.67
	IR-TWK	635.66	139.20	162.83	67.67
10	TWK-FB	560.47	81.09	119.59	56.00
	IR-TWK	634.19	141.19	166.07	67.33

FINAL ANSWER

	MEAN FLOW TIME	STANDARD DEVIATION FLOW TIME	MEAN TARDINESS	MEAN ABSOLUTE LATENESS	% of TARDY JOBS
TWK-FB	561.70	10.17	79.98	119.80	55.30
IR-TWK	637.24	7.07	140.37	165.04	67.13

PRODUCTS E, P and F
Due_date factor: 1.5 Overlapping Level % 0.90

r	RULE	MEAN FLOW TIME	MEAN TARDINESS	MEAN ABSOLUTE LATENESS	% of TARDY JOBS
1	TWK-FB	787.58	144.23	205.74	64.67
	IR-TWK	831.35	171.32	216.14	69.67
2	TWK-FB	788.84	140.93	189.61	69.00
	IR-TWK	811.38	160.65	206.51	69.00
3	TWK-FB	772.66	131.88	191.89	62.67
	IR-TWK	823.36	165.88	209.20	71.33
4	TWK-FB	762.75	130.43	194.36	64.33
	IR-TWK	818.72	169.11	215.75	68.33
5	TWK-FB	770.33	133.94	197.89	62.00
	IR-TWK	819.06	166.14	213.56	66.33
6	TWK-FB	778.96	138.53	199.04	65.33
	IR-TWK	827.85	169.46	212.00	70.33
7	TWK-FB	765.83	134.46	201.18	61.67
	IR-TWK	819.72	168.45	215.29	68.67
8	TWK-FB	783.20	146.57	199.58	66.67
	IR-TWK	823.90	174.83	215.40	69.67
9	TWK-FB	761.71	132.78	198.11	64.33
	IR-TWK	813.13	165.11	211.25	66.33
10	TWK-FB	768.13	137.12	200.75	63.33
	IR-TWK	817.91	172.75	214.57	70.67

FINAL ANSWER

	MEAN FLOW TIME	STANDARD DEVIATION FLOW TIME	MEAN TARDINESS	MEAN ABSOLUTE LATENESS	% of TARDY JOBS
TWK-FB	774.00	10.04	137.09	197.82	64.40
IR-TWK	820.64	6.16	168.37	212.97	69.03

PRODUCTs A, L, B, M and G
Due_date factor: 2.5 Overlapping Level % 0.90

r	RULE	MEAN FLOW TIME	MEAN TARDINESS	MEAN ABSOLUTE LATENESS	% of TARDY JOBS
1	TWK-FB	546.71	236.91	248.49	79.40
	IR-TWK	574.62	264.51	275.78	78.20
2	TWK-FB	543.60	232.26	242.78	79.60
	IR-TWK	560.95	249.96	260.83	78.20
3	TWK-FB	544.75	240.56	252.42	81.40
	IR-TWK	563.08	256.69	266.35	79.40
4	TWK-FB	548.83	239.98	251.82	79.00
	IR-TWK	575.37	263.66	272.66	79.60
5	TWK-FB	554.77	245.82	255.76	81.80
	IR-TWK	569.44	258.91	267.27	80.00
6	TWK-FB	544.90	237.14	249.07	79.80
	IR-TWK	561.80	252.87	263.64	79.40
7	TWK-FB	541.34	234.29	246.36	79.00
	IR-TWK	558.57	250.59	261.73	76.60
8	TWK-FB	546.57	239.82	251.49	81.40
	IR-TWK	557.50	248.64	258.19	77.80
9	TWK-FB	542.57	234.80	247.58	78.40
	IR-TWK	558.18	247.19	256.75	79.80
10	TWK-FB	542.85	235.62	245.66	80.40
	IR-TWK	558.66	251.48	261.58	78.40

FINAL ANSWER

	MEAN FLOW TIME	STANDARD DEVIATION FLOW TIME	MEAN TARDINESS	MEAN ABSOLUTE LATENESS	% of TARDY JOBS
TWK-FB	545.69	3.90	237.72	249.14	80.02
IR-TWK	563.82	6.83	254.45	264.48	78.74

PRODUCTS C, J, D, K and H
Due_date factor: 2.5 Overlapping Level % 0.90

r	RULE	MEAN FLOW TIME	MEAN TARDINESS	MEAN ABSOLUTE LATENESS	% of TARDY JOBS
1	TWK-FB	659.09	31.73	189.55	26.40
	IR-TWK	721.23	63.35	190.67	36.80
2	TWK-FB	657.23	26.49	175.27	23.40
	IR-TWK	704.56	50.75	176.46	34.40
3	TWK-FB	651.04	24.41	175.65	23.20
	IR-TWK	711.97	60.36	186.64	37.80
4	TWK-FB	656.42	30.24	186.50	23.00
	IR-TWK	717.15	58.42	182.14	34.60
5	TWK-FB	647.80	23.52	178.59	20.80
	IR-TWK	705.34	54.27	182.55	32.60
6	TWK-FB	648.45	24.13	175.32	22.20
	IR-TWK	708.08	56.62	180.68	35.80
7	TWK-FB	656.02	28.39	187.86	24.40
	IR-TWK	722.86	61.16	186.57	35.80
8	TWK-FB	647.68	24.61	189.11	22.00
	IR-TWK	716.62	54.35	179.65	35.80
9	TWK-FB	652.42	26.46	180.23	23.20
	IR-TWK	715.54	58.70	181.58	38.20
10	TWK-FB	646.27	21.36	177.30	21.00
	IR-TWK	695.18	43.95	173.56	30.60

FINAL ANSWER

	MEAN FLOW TIME	STANDARD DEVIATION FLOW TIME	MEAN TARDINESS	MEAN ABSOLUTE LATENESS	% of TARDY JOBS
TWK-FB	652.24	4.66	26.13	181.54	22.96
IR-TWK	711.85	8.57	56.19	182.05	35.24

APPENDIX 9: SCHEDULING REPORTS

PRODUCTS G,L and M

OPERATION	MACHINE	START TIME	FINISH TIME
G16	a	44.78	67.48
L9	a	67.48	95.36
G17	a	720.86	742.65
G7	b	0.00	38.32
G19	b	38.32	74.11
G21	b	74.11	108.60
L7	b	108.60	139.86
L6	b	139.86	176.27
G10	b	176.27	209.81
M13	b	209.81	239.67
M9	b	239.67	278.58
M16	b	278.58	309.55
L1	b	330.38	369.42
G15	b	378.49	416.42
G4	b	416.42	449.63
M1	b	477.82	515.67
G22	c	0.00	40.80
M2	c	200.06	247.36
M7	c	259.63	302.19
M5	c	407.98	454.99
G11	c	557.65	598.77
G20	c	696.40	742.65
G9	d	209.81	269.13
M8	d	278.58	337.14
M15	d	345.28	398.65
M14	d	398.65	449.63
G12	d	498.78	557.65
G18	d	568.09	620.41
G2	d	620.41	672.25
L10	e	0.00	67.48
L4	e	67.48	133.00
M3	e	133.00	200.06
M12	e	313.88	379.80
G13	e	429.98	498.78
G3	e	553.77	620.41
G1	e	770.83	831.33
M4	f	0.00	76.59
M10	f	76.59	148.08
L5	f	176.27	252.62
L8	f	252.62	330.38
M6	f	330.38	407.98
M11	f	407.98	477.82
G6	f	477.82	548.08
G5	f	548.08	626.95
G8	f	626.95	700.44

G14	f	700.44	770.83

PRODUCTS H, J and K

OPERATION	MACHINE	START TIME	FINISH TIME
H8	a	103.53	127.33
J8	a	127.33	155.21
H18	a	297.49	319.28
J2	a	319.28	344.24
K16	b	0.00	30.98
H16	b	30.98	68.90
H22	b	68.90	103.39
J6	b	103.39	134.65
J5	b	134.65	171.06
H7	b	171.06	209.38
H20	b	209.38	245.17
K10	b	360.86	396.41
H4	b	429.86	463.08
H10	b	463.08	496.62
J1	b	496.62	535.66
H17	b	535.66	572.38
H23	c	28.11	68.90
H12	c	117.27	158.39
K13	c	158.39	202.90
K6	c	412.99	455.55
H21	c	470.51	516.76
K1	c	586.32	633.62
J10	d	0.00	54.04
K15	d	54.04	107.41
K14	d	107.41	158.39
K4	d	158.39	214.44
H19	d	245.17	297.49
K8	d	297.49	356.05
K7	d	356.05	412.99
H11	d	412.99	463.08
H9	d	641.19	700.51
H2	d	727.53	779.38
J7	e	0.00	59.85
J9	e	59.85	127.33
K12	e	127.33	193.25
J3	e	193.25	258.77
K2	e	421.49	488.54
H14	e	488.54	553.48
H13	e	572.38	641.19
H3	e	660.90	727.53
H1	e	779.38	839.87
H15	f	68.90	139.30
K3	f	214.44	291.03
K11	f	291.03	360.86
J4	f	360.86	437.22

K9	f	437.22	508.71
K5	f	508.71	586.32
H6	f	611.34	681.60
H5	f	700.51	779.38

PRODUCTS A,B,L,M and G

OPERATION	MACHINE	START TIME	FINISH TIME
L10	a	0.00	22.27
G12	a	329.51	357.38
B9	a	357.38	379.18
B8	a	379.18	401.88
M10	a	463.14	489.55
G6	a	712.46	737.42
G10	b	0.00	31.26
L9	b	31.26	66.34
A6	b	66.34	104.66
A4	b	104.66	138.32
G3	b	138.32	172.81
G9	b	184.25	220.67
G22	b	220.67	259.57
L1	b	263.58	299.12
B3	b	299.12	332.67
M4	b	428.19	464.98
M3	b	464.98	499.54
M14	b	499.54	535.93
G5	b	737.42	776.46
G14	b	904.69	942.55
G4	c	97.53	138.32
G2	c	172.81	219.06
L5	c	219.06	263.58
M13	c	364.80	410.66
G18	c	410.66	457.67
G15	c	857.39	904.69
L7	d	0.00	53.36
B5	d	53.36	112.23
A2	d	112.23	164.07
L6	d	164.07	215.06
B4	d	215.06	265.14
B10	d	265.14	317.46
M16	d	317.46	373.03
B2	d	373.03	432.35
M6	d	432.35	489.55
M9	d	489.55	546.06
M8	d	546.06	599.97
M1	d	656.91	716.23
G17	d	716.23	772.28
G21	d	827.05	885.61
G20	d	885.61	942.55
G1	d	942.55	997.32

A3	e	0.00	66.64
L8	e	66.64	135.62
L4	e	135.62	201.53
A1	e	201.53	262.03
G13	e	262.03	329.51
B6	e	329.51	394.45
M15	e	394.45	458.28
M12	e	458.28	522.30
M11	e	522.30	583.54
M2	e	583.54	646.94
G7	e	646.94	712.46
G16	e	790.33	857.39
M7	f	0.00	71.31
A5	f	104.66	174.93
B7	f	207.06	277.45
G19	f	277.45	355.06
B1	f	432.35	505.83
M5	f	505.83	580.29
G11	f	663.13	740.88
G8	f	866.19	942.55

PRODUCTS C,D,H,J and K

OPERATION	MACHINE	START TIME	FINISH TIME
D8	a	0.00	22.71
H4	a	22.71	47.66
J7	a	47.66	69.94
D9	a	385.29	407.08
K7	a	764.44	790.84
H10	a	942.01	969.89
K1	a	984.96	1011.87
J10	b	0.00	34.56
H8	b	34.56	65.82
J6	b	69.94	105.02
C7	b	105.02	143.34
D7	b	143.34	181.26
H7	b	200.53	236.94
K12	b	236.94	273.32
J4	b	273.32	304.30
J1	b	304.30	334.16
C4	b	334.16	367.82
H3	b	419.85	458.89
H2	b	458.89	493.25
D1	b	493.25	526.80
H22	b	526.80	562.34
D3	c	111.18	152.30
D12	c	152.30	198.55
J2	c	214.01	258.52
K10	c	626.99	672.86
H18	c	1100.46	1143.02

H13	c	1220.63	1267.93
H1	c	1403.65	1444.45
D10	d	0.00	52.32
D4	d	52.32	111.18
C2	d	111.18	163.03
J3	d	163.03	214.01
J8	d	214.01	273.32
K3	d	273.32	330.52
D11	d	330.52	385.29
K11	d	385.29	440.39
D2	d	440.39	490.48
K16	d	490.48	547.00
K15	d	547.00	600.19
K14	d	600.19	659.06
K13	d	659.06	714.63
K8	d	714.63	764.44
H16	d	764.44	820.49
H12	d	820.49	874.53
K6	d	874.53	931.04
K5	d	931.04	984.96
H20	d	984.96	1043.52
H19	d	1043.52	1100.46
H5	d	1345.68	1403.65
C3	e	0.00	66.64
J9	e	66.64	130.03
J5	e	130.03	199.02
C1	e	367.82	428.32
D5	e	428.32	493.25
K9	e	714.63	778.64
H11	e	874.53	942.01
H14	e	942.01	1009.07
K4	f	0.00	71.31
C8	f	71.31	144.79
C6	f	144.79	215.06
C5	f	215.06	293.93
D6	f	293.93	364.33
H23	f	364.33	434.16
H21	f	562.34	633.84
K2	f	734.79	809.25
H15	f	820.49	897.07
H6	f	1066.67	1143.02
H17	f	1143.02	1220.63
H9	f	1267.93	1345.68

PRODUCTS E,F and P

OPERATION	MACHINE	START TIME	FINISH TIME
P6	a	121.13	148.04
E7	a	148.04	171.85
F9	a	450.70	475.66

F39	a	892.98	919.61
F38	a	919.61	941.89
F4	a	941.89	963.68
F17	a	1170.60	1198.47
F2	a	1287.51	1310.22
P14	b	0.00	36.38
E8	b	36.38	70.04
E4	b	70.04	103.26
F33	b	137.89	167.75
P3	b	167.75	204.54
E10	b	204.54	238.08
F14	b	250.05	281.31
E6	b	281.31	319.63
F8	b	475.66	514.70
F7	b	514.70	549.19
P20	b	549.19	579.52
F13	b	589.28	623.65
P17	b	623.65	660.04
F22	b	666.95	704.81
F36	b	722.50	753.48
P2	b	753.48	788.04
F37	b	1013.99	1049.07
F30	b	1049.07	1084.62
F28	b	1156.12	1191.90
F3	b	1250.79	1287.51
F1	b	1310.22	1348.14
F24	c	40.04	87.05
E11	c	87.05	128.17
F10	c	610.90	651.70
F6	c	651.70	697.95
F25	c	944.73	992.03
E13	d	0.00	58.87
P11	d	58.87	113.97
P4	d	113.97	171.17
P15	d	171.17	230.04
P12	d	230.04	285.61
E2	d	285.61	337.45
P9	d	337.45	387.26
P19	d	387.26	442.50
P8	d	442.50	499.01
P5	d	499.01	552.93
F11	d	552.93	610.90
F23	d	610.90	666.95
F5	d	717.93	772.70
P16	d	772.70	829.23
F27	d	829.23	887.79
F26	d	887.79	944.73
F35	d	944.73	998.09
F34	d	998.09	1049.07
F19	d	1049.07	1103.12
F16	d	1198.47	1250.79
P13	e	0.00	64.01
E12	e	64.01	128.95

P10	e	128.95	190.20
F15	e	190.20	250.05
F32	e	250.05	315.96
E3	e	319.63	386.26
E1	e	386.26	446.76
P18	e	446.76	513.76
P1	e	829.23	892.62
F20	e	892.62	959.68
F18	e	1103.12	1170.60
P7	f	0.00	71.31
E9	f	71.31	144.79
E5	f	144.79	215.06
P21	f	215.06	293.19
P22	f	293.19	365.32
F21	f	365.32	441.91
F12	f	441.91	518.26
F31	f	518.26	588.10
F29	f	1084.62	1156.12

PRODUCTS A,B,C,D,E,F,G,H,J,K,L,M,N and P

OPERATION	MACHINE	START TIME	FINISH TIME
J7	a	0.00	23.12
H20	a	23.12	47.12
G4	a	47.12	69.98
D4	a	70.47	98.34
H11	a	133.86	159.76
H10	a	159.76	183.10
B8	a	345.67	368.38
H15	a	368.38	389.63
B9	a	389.63	411.42
C6	a	514.29	539.24
J2	a	577.85	604.54
E11	a	604.54	631.17
F24	a	872.64	896.57
F23	a	896.57	925.42
P5	a	925.42	954.48
N12	a	954.48	980.84
H21	a	996.77	1020.12
G19	a	1049.40	1078.53
H17	a	1137.59	1165.68
N10	a	1386.06	1407.14
N11	a	1490.41	1513.54
K9	a	1513.54	1533.98
H6	a	1543.78	1564.86
H13	a	1564.86	1588.89
M1	a	1783.59	1809.89
F38	a	1878.27	1901.64
F6	a	2650.43	2676.83
F1	a	2970.22	2997.13

G10	b	0.00	37.47
J6	b	37.47	68.09
A6	b	68.09	106.41
J10	b	106.41	142.17
A4	b	142.17	175.84
P13	b	203.53	234.96
C3	b	234.96	269.45
G7	b	296.96	329.91
H16	b	329.91	366.05
E9	b	366.05	397.03
E12	b	397.03	431.59
F36	b	431.59	468.58
G16	b	468.58	506.24
G15	b	506.24	544.92
C5	b	544.92	583.96
K6	b	583.96	622.60
J4	b	622.60	652.21
C4	b	652.21	686.58
J1	b	686.58	725.64
L9	b	725.64	762.56
B3	b	762.56	796.10
E10	b	796.10	831.18
D6	b	831.18	869.04
H8	b	869.04	899.62
P11	b	906.34	944.33
M3	b	944.33	974.50
P12	b	1021.22	1060.07
N13	b	1060.07	1099.05
H18	b	1099.05	1137.59
N3	b	1137.59	1170.49
D1	b	1225.09	1256.35
E3	b	1276.08	1311.63
E1	b	1405.63	1444.54
H9	b	1588.89	1620.68
G21	b	1657.51	1690.89
P17	b	1826.19	1857.44
P16	b	1857.44	1888.50
H2	b	1942.55	1978.99
F26	b	2424.76	2459.64
F18	b	2525.35	2558.31
F17	b	2558.31	2588.64
N18	c	0.00	41.35
N16	c	41.35	88.59
H12	c	88.59	133.86
G9	c	198.01	241.88
N5	c	241.88	289.39
N15	c	289.39	333.51
K11	c	333.51	374.71
C2	c	374.71	420.96
L4	c	420.96	464.47
L7	c	464.47	506.71
D11	c	506.71	549.27
G6	c	614.97	660.65

M12	c	660.65	700.49
D7	c	700.49	747.80
D10	c	747.80	794.80
G5	c	809.40	855.52
H22	c	855.52	900.79
E5	c	900.79	945.30
M9	c	945.30	990.25
P21	c	1043.59	1091.21
N4	c	1091.21	1133.79
K15	c	1332.45	1373.33
K14	c	1373.33	1413.10
M15	c	1440.63	1480.39
F33	c	1556.32	1601.98
G17	c	1601.98	1648.65
G20	c	1722.92	1766.97
G12	c	1766.97	1809.44
G11	c	1809.44	1852.64
F31	c	1929.93	1974.18
F8	c	2039.02	2084.89
P3	c	2084.89	2125.48
F35	c	2424.61	2467.72
F16	c	2649.22	2690.13
B4	d	0.00	50.09
B5	d	50.09	108.95
A2	d	108.95	160.80
J9	d	160.80	217.15
D12	d	217.15	274.09
K12	d	274.09	333.51
B10	d	333.51	385.83
E13	d	385.83	445.14
K7	d	445.14	498.02
E7	d	498.02	551.38
M7	d	551.38	610.11
M13	d	610.11	660.65
G8	d	660.65	719.06
C1	d	719.06	773.83
L5	d	773.83	833.19
M5	d	833.19	887.87
M4	d	887.87	944.33
N17	d	944.33	1001.01
L1	d	1001.01	1055.68
B2	d	1055.68	1115.00
D9	d	1115.00	1171.05
D5	d	1171.05	1225.09
E6	d	1225.09	1276.08
K16	d	1276.08	1332.45
K10	d	1332.45	1386.65
M16	d	1386.65	1440.63
N14	d	1440.63	1490.41
H7	d	1490.41	1543.78
G18	d	1543.78	1601.98
G22	d	1601.98	1657.51
G13	d	1657.51	1715.47

P10	d	1715.47	1772.76
P19	d	1772.76	1826.21
F39	d	1826.21	1878.27
F32	d	1878.27	1929.93
P6	d	1929.93	1983.92
F9	d	1983.92	2039.02
F12	d	2039.02	2094.59
K1	d	2094.59	2152.64
F15	d	2152.64	2209.16
F14	d	2209.16	2262.34
H1	d	2262.34	2312.68
F20	d	2312.68	2367.92
F27	d	2367.92	2424.76
F7	d	2424.76	2474.57
F34	d	2474.57	2528.25
F5	d	2676.83	2733.34
F29	d	2733.34	2784.65
F28	d	2784.65	2838.57
F3	d	2838.57	2895.77
A3	e	0.00	66.64
H4	e	66.64	130.43
H23	e	130.43	191.91
K8	e	191.91	261.38
D2	e	261.38	321.22
D8	e	321.22	388.28
A1	e	388.28	448.77
C7	e	448.77	514.29
J3	e	514.29	577.85
K4	e	577.85	639.11
L6	e	639.11	701.85
H14	e	701.85	767.56
M11	e	767.56	832.95
E8	e	832.95	898.87
L8	e	898.87	958.52
P15	e	958.52	1021.22
M2	e	1021.22	1084.60
B6	e	1084.60	1149.54
G3	e	1149.54	1214.02
G2	e	1214.02	1281.15
P22	e	1281.15	1350.64
N6	e	1350.64	1417.44
K13	e	1417.44	1479.52
P18	e	1479.52	1545.14
K2	e	1559.07	1625.59
N7	e	1625.59	1691.39
P7	e	1691.39	1759.69
P20	e	1759.69	1826.19
P4	e	2037.69	2103.73
N1	e	2126.85	2195.07
P2	e	2195.07	2263.86
F13	e	2263.86	2327.87
F19	e	2367.92	2434.93
F25	e	2459.64	2525.35

F11	e	2525.35	2589.18
F10	e	2589.18	2650.43
F30	e	2650.43	2713.95
L10	f	0.00	75.97
P8	f	75.97	147.98
D3	f	147.98	225.73
A5	f	225.73	296.00
J5	f	296.00	371.16
C8	f	371.16	447.51
J8	f	447.51	518.28
M10	f	518.28	596.87
H19	f	596.87	675.34
M6	f	675.34	749.25
B7	f	749.25	819.64
P14	f	819.64	893.45
G14	f	893.45	964.81
E4	f	964.81	1034.64
N9	f	1034.64	1110.40
F22	f	1110.40	1188.52
F21	f	1188.52	1260.65
B1	f	1260.65	1334.14
E2	f	1334.14	1405.63
M8	f	1405.63	1482.42
K3	f	1482.42	1559.07
H3	f	1559.07	1637.73
N8	f	1637.73	1707.52
M14	f	1707.52	1783.59
P9	f	1783.59	1858.45
F37	f	1901.64	1972.60
K5	f	1972.60	2048.23
N2	f	2048.23	2126.85
H5	f	2126.85	2204.00
G1	f	2204.00	2277.38
P1	f	2277.38	2355.47
F4	f	2733.34	2804.65
F2	f	2895.77	2970.22