

THE UNIVERSITY OF HULL

A Pattern Enumeration Approach
to the Trim Loss Problem

being a Thesis submitted for the Degree of
Doctor of Philosophy in the University of Hull

by

IAN LAWSON COVERDALE, M.A.(Cantab), M.Sc.

September 1978.

BEST COPY

AVAILABLE

Poor text in the original
thesis.

Some text bound close to
the spine.

SUMMARY

This thesis examines the characteristics of practical one-dimensional trim loss problems. As a result of the wide range of these characteristics, previous scheduling methods have only had a limited range of applicability. A heuristic approach is proposed, based on pattern enumeration, which can be used to develop scheduling methods for a reasonably wide class of trim loss problems. The effectiveness of the approach depends on its ability to avoid the intractable residual problems which normally arise towards the end of a heuristic scheduling procedure. The approach is used in three case studies, and the efficiency of the schedules generated is compared with that yielded by other methods.

CONTENTS

	<u>Page</u>
1. Outline of the Problem	
1.1 Introduction	1
1.2 Roll Slitting	2
1.3 Bar Cropping	3
1.4 Glossary of Terms	3
1.5 The Trim Loss Problem in Practice	5
1.6 Mathematical Formulation	6
2. Review of Current Methods	
2.1 Introduction	10
2.2 Manual Methods	10
2.3 Heuristic Methods	12
2.4 Branch and Bound Techniques	14
2.5 Linear Programming	16
3. Limitations of Current Methods	
3.1 Introduction	18
3.2 Throughput Constraints and Set-up Costs	18
3.3 Integer Problems.	20
3.4 Under and Over Production	21
3.5 Summary	21
4. The Pattern Enumeration Technique	
4.1 Introduction	23
4.2 The Basic Pattern Enumeration Method	24
4.3 A Numerical Example	28
4.4 Pattern Enumeration Techniques	36
4.5 Computational Efficiency	40
4.6 Introduction to the Case Studies	43
5. A Roll Slitting Problem	
5.1 The Scheduling Problem	45
5.2 Detailed Problem Characteristics	46
5.3 Mathematical Formulation	48
5.4 Current Scheduling Method	48
5.5 Proposed Method	51
5.6 Discussion	58

6.	A Bar Cutting Problem	
6.1	The Scheduling Problem	60
6.2	Detailed Characteristics	61
6.3	Mathematical Formulation	64
6.4	A Mathematical Programming Method	65
6.5	Current Scheduling Method	70
6.6	Proposed Method	72
6.7	Subsequent Experiments	80
6.8	Discussion	82
7.	A Sequential Bar Cutting Problem	
7.1	The Scheduling Problem	84
7.2	Detailed Characteristics	86
7.3	Current Scheduling Method	87
7.4	Proposed Method	90
7.5	Discussion	100
8.	Conclusions	
8.1	Discussion of the Case Studies	102
8.2	Evaluation of the Pattern Enumeration Approach	104
8.3	The Application of Pattern Enumeration Methods	107
8.4	Suggestions for further research	108
8.5	Summary	109
Appendix 1. Enumeration Routines		
1.1	Data	111
1.2	Programs	111
Appendix 2. The Roll Slitting Problem		
2.1	Data	116
2.2	Programs	125
2.3	Results	129
Appendix 3. The Bar Cutting Problem		
3.1	Data	134
3.2	Programs	138
3.3	Results	155
Appendix 4. The Sequential Bar Cutting Problem		
4.1	Data	158
4.2	Programs	175
4.3	Results	181
Appendix 5. Bibliography.		
		188

1. OUTLINE OF THE PROBLEM

1.1 Introduction

Many industries produce or buy material in much larger sizes than those required for assembly into finished products; the material is then cut to the sizes required. By producing or buying large quantities of material in standard sizes the cost of materials is diminished. An additional advantage is that stocks can be reduced, as all requirements can be met from only a limited number of stock sizes.

Against these reductions in cost must be set the cost of the operations involved in cutting the stock sizes of material to the sizes required. The problem for the production scheduler is to choose which of the small pieces are to be cut from any particular large piece of the basic material. It is almost inevitable that the small pieces will not fit exactly into the large piece but that there will be a certain amount of unusable material (or trim) left over. It is the task of the scheduler to minimize the total production costs of the cutting operations, one of the important costs being that of the material wasted as trim. Some estimate of the economic significance of the problem can be made when one considers that most industries facing it regard trim losses in the range 1% - 5% of the total material produced to be acceptable. When one applies this percentage loss over the wide range of industries facing the problem, it will be realized that even a slight reduction in waste is of great value.

Problems of this kind arise in the paper industry [(11), (12), (13), (15), (16), (17), (45), (50), (52) & (55)] and the metal processing industries [(3), (7), (8), (44), & (56)]. They also occur to a lesser extent in the glass industry [(33) & (34)] and the timber industry.

The question of which sizes to stock is not considered in this thesis. The scope of the study has been restricted to one particular class of the scheduling problems outlined above, known as the one-dimensional trim problem (this term is used to signify that the cutting operations considered take place in only one plane).

There are two main situations in which one-dimensional trim problems occur:

- (a) in the slitting of rolls;
- (b) in the bar cropping process.

This class of problems may seem very restrictive but despite its relatively simple basic formulation, outlined below, it does include the majority of situations where material is to be cut.

1.2 Roll Slitting

In both the paper and the steel coil industries rolls of material need to be slit. Material is produced in large width rolls which are then cut to those finally required by unrolling them over a series of parallel cutting edges and re-rolling the slit coils separately (as shown in Fig. 1 below).

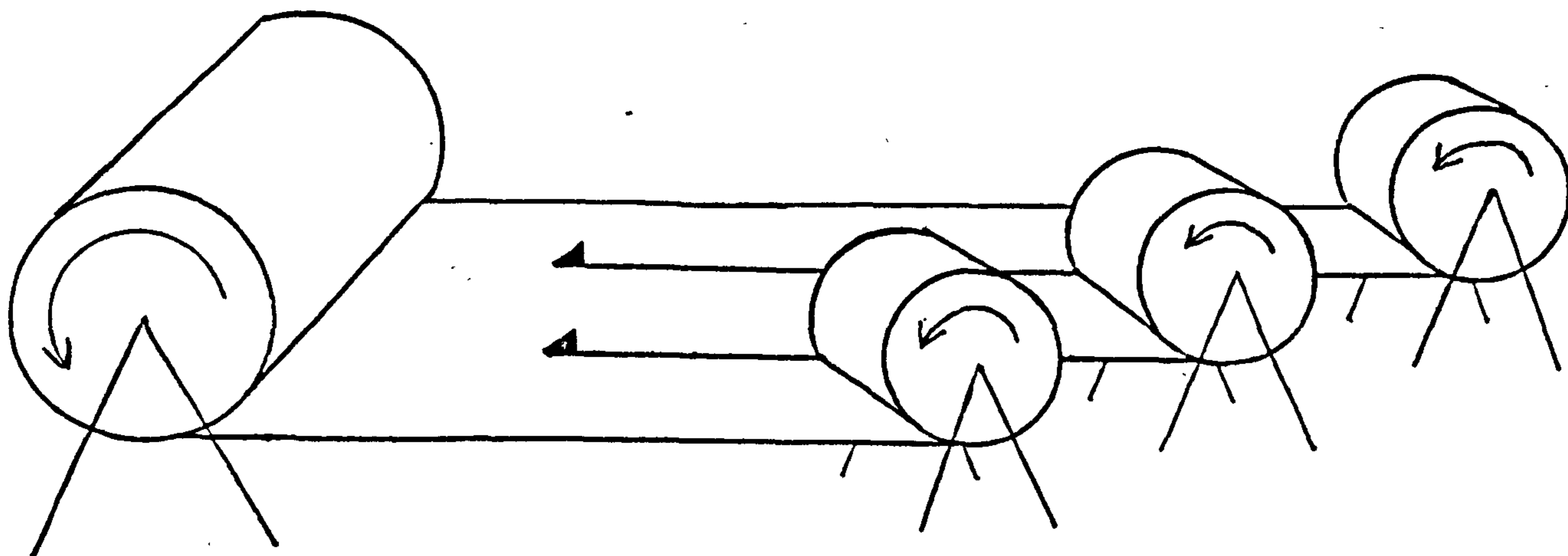


Figure 1 The Roll Slitting Process

The production scheduling problem for this process is a one-dimensional trim problem as the only way the roll can be cut is along its length (cuts perpendicular to this are regarded as terminating a particular cutting pattern). In the paper industry in particular, problems of this type present very great scheduling difficulties as a large width roll of up to 20 ft. will have to be cut into sizes ranging from only a few inches to 10 ft.

1.3 Bar Cropping

Bar cropping, while physically a far simpler operation than roll slitting, presents a very similar scheduling problem. In bar cropping operations, long lengths of material (e.g. planks of wood or steel girders) are cut, by a shearing or sawing operation, to the required lengths.

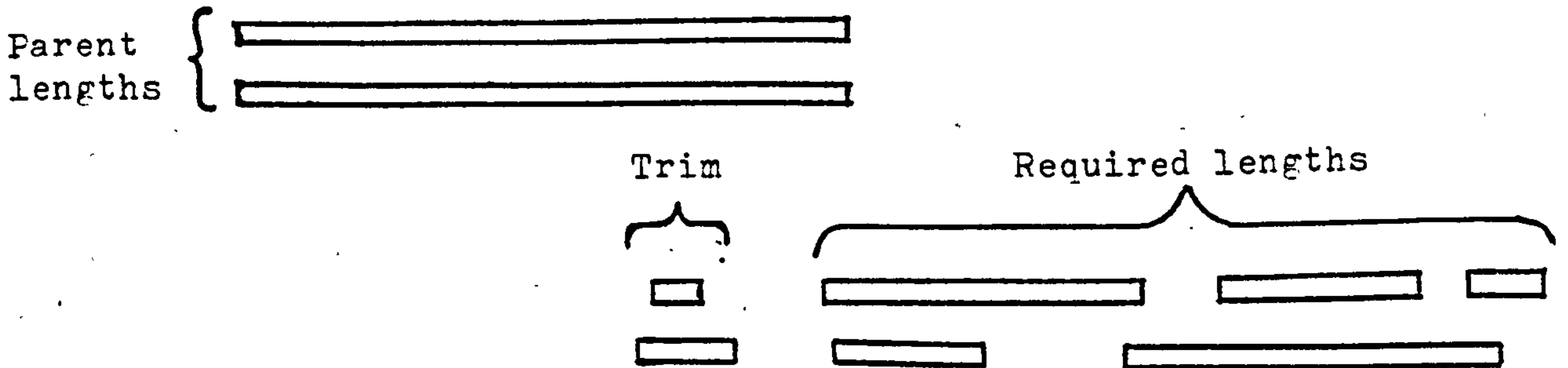


Figure 2 The Bar Cropping Process

The scheduling problem is, as for roll slitting, to choose which of the smaller sizes are to be cut from a particular large size. The case study of Chapter 6 is a typical example of the scheduling problem incurred in a bar cropping operation. In that particular study, the product under consideration is steel rods for use in reinforced concrete.

1.4 Glossary of Terms

Size, throughout the thesis, refers to the length of the material perpendicular to the cutting plane. In the case of a roll to be slit, this is the width, and with a bar, the length.

Quantity refers to the amount of material to be cut. In the case of a roll to be slit, this is expressed as either the length of unrolled material, the number of rolls, or the weight of material, while in the case of bar cropping it is the number of bars,

With this definition of size and quantity, a single mathematical formulation can be applied to both roll slitting and bar cropping scheduling problems (see § 1.6, p. 6).

Parent rolls/bars are the large pieces of material from which all requirements must be cut.

Order rolls/bars are the pieces of material which are required to be cut from the parent rolls/bars to satisfy demand. Order sizes, order

quantities and order list (the list of all order sizes with the appropriate order quantities) are similarly defined.

An Off-cut is material, cut from a parent size, which does not satisfy a current order. Off-cuts are produced to fill out the parent roll/bar which is being cut, and hence reduce waste. The off-cut, when produced, must be held in stock until it can be either dispatched to meet a new order or treated as a parent size and recut (e.g. $\phi 6.2$ p.63 and $\phi 7.2$ p.86).

The Trim is scrap material which is too small to be worth saving in stock as orders will normally be for sizes larger than this. In situations where off-cuts may not be held, all material which can not be used to satisfy current orders must be regarded as trim.

A Pattern is a combination of sizes to be cut from a parent size. It may include order sizes, off-cuts and trim, e.g. with a parent size of 13" and a demand for 5" and 3" sizes, then one piece of 5" and two of 3" with a trim of 2" would be a pattern.

A Schedule is a set of patterns such that the sizes produced, when parent lengths are cut by this set of patterns, meet the current order list.

A Feasible Solution to a production scheduling problem is any schedule of patterns which would meet the demand for all sizes ordered without violating any of the additional restrictions of that particular problem (e.g. a limit on the number of patterns to be used). The term 'feasible pattern' is used to describe a pattern which can in practice be used to cut a parent size.

The Global Optimum Solution (or the Optimum Solution) to a production scheduling problem is the best, in some clearly defined sense, (e.g. minimum cost). of all the feasible solutions.

An Algorithm refers, in this thesis, to "a series of operations to be performed on the data of a production scheduling problem which, in a finite number of steps, will give the global optimum solution if one exists" (Thompson p. 135 in (17)) e.g. the linear programming method for the simple trim loss problem.

A Heuristic Procedure, in contrast to an algorithm, is "a series of operations on the data of a production scheduling problem which, in a finite number of steps, will give a feasible solution if one exists" (Thompson p. 137 in (17)). This feasible solution will normally be

acceptable in terms of the scheduler's objectives. Whereas an algorithm will be supported by mathematical theory, the justification for a heuristic is simply that it appears to work.

1.5 The Trim Loss Problem in Practice

The classic formulation of the trim loss problem is merely to minimize the trim loss incurred in meeting a set of orders. However the problem is rarely encountered in this simple format in practice. Other constraints and costs have to be considered which complicate the formulation:

(a) Production constraints other than that of meeting demand:

(i) Restrictions on the number of pieces which can be cut in a pattern. This constraint would arise, for example, if there were a limited number of cutting edges on a slitting machine;

(ii) Constraints caused by the characteristics of the different cutting machines, e.g. only certain sizes or qualities can be cut on a particular machine;

(iii) Sometimes there is a need to produce an integer solution. This is an essential characteristic of bar cropping problems, and frequently in roll slitting problems the number of rolls cut by a pattern must be a whole number. (This is often related to high costs of changing patterns as in (b) (i) below). The need for integer solutions also arises when joins are not allowed in a finished roll.

(b) Production costs other than the cost of trim:

(i) Frequently, particularly in roll slitting, there will be a lost opportunity cost arising from the time lost in changing the setting of a machine to cut a different pattern. In some cases the set-up time between patterns can be one or two hours (the equivalent of 25% of production time). In this situation the scheduler will obviously attempt to use as few setting changes as possible. This can be done by attaching a cost to every set-up used, and minimizing the overall cost of trim and set-ups. Alternatively a restriction on the number of permissible set-ups can be added to the constraints.

(c.f. goal programming);

- (ii) Some production schedules may incur extra costs after the cutting stage as a result of the particular cutting schedule used. These additional cutting costs are more likely to occur in complex roll slitting problems. In particular, costs will arise if there are restrictions on the shape of the finished rolls (e.g. a certain inside or outside diameter, in which case if demands for sizes requiring a different diameter are scheduled together in one pattern, it will be necessary to re-roll at least one of them at a later stage at extra cost).
- (c) In many practical situations the cost of any of the schedules satisfying the problem constraints is excessive. In these situations the scheduler may have the option of weakening some of the problem constraints:
- (i) By overfilling or underfilling a customer's order (in many situations there is an accepted tolerance of 5 - 10%, e.g. §5.2 p.47). This extra freedom frequently helps the scheduler to eliminate wasteful patterns;
 - (ii) By producing "cut sizes" to stock. This can be done either to fully utilize the cutting machines or, as (i) above, to give greater freedom in choosing patterns. If this option is used it does, of course, raise the corresponding problem later as to when material should be taken from stock and possibly be recut to satisfy demand;
 - (iii) By having several parent sizes from which orders may be cut.

Fortunately not all the complicating factors outlined above will be present in any particular situation.

1.6 Mathematical Formulation

The classic trim loss problem (i.e. demand to be met and trim minimized) can be formulated in a linear programming form.

Define W to be the parent size and d_j the quantity demanded of size w_j ($j = 1, 2, \dots, m$)

The i^{th} pattern of n (the total number of possible patterns) is

defined by

$$\begin{pmatrix} a_{i1} \\ a_{i2} \\ a_{i3} \\ \vdots \\ a_{im} \end{pmatrix} \quad a_{ij} \text{ is integer } \quad \forall i \quad \forall j$$

where a_{ij} is the number of pieces of size w_j to be cut by the i^{th} pattern.

Beged-Dov (1) gives a proof that

$$\frac{W^m}{m! \prod_{j=1}^m w_j} \leq n \leq \frac{\left(W + \sum_{j=1}^m w_j\right)^m}{m! \prod_{j=1}^m w_j}$$

(The magnitude of n is considered in §2.5, p.16)

Let t_i be the trim from pattern i , then

$$W = t_i + \sum_{j=1}^m a_{ij} w_j \quad i = 1, 2, \dots, n$$

Define x_i to be the quantity (i.e. the length of rolled material if slitting rolls, or number of bars if cropping bars) to be cut by pattern i .

Then we require

$$x_i \geq 0 \quad i = 1, 2, \dots, n$$

$$\sum_{i=1}^n a_{ij} x_i = d_j \quad j = 1, 2, \dots, m$$

Minimize

$$Z = \sum_{i=1}^n t_i x_i$$

(or equivalently minimize $Z' = \sum_{i=1}^n x_i$)

Most of the complexities outlined in (a)-(c) of §1.5 can be incorporated into this formulation but they destroy its linear nature.

(a) Production constraints other than that of meeting demand:

(i) Limited number of pieces (N) in a pattern:

$$\sum_{j=1}^m a_{ij} \leq N \quad \forall i$$

This can easily be introduced by restricting the set of feasible patterns to only those containing less than or equal to the required number of pieces.

(ii) Machine restrictions:

This is a very difficult restriction to include in the mathematical formulation and can only be done by dividing the set of feasible patterns into subsets S_1, S_2, \dots, S_K such that S_k is the set of patterns which can be cut on machine k , and adding the extra constraints:

$$\sum_{x_i \in S_k} x_i \leq C_k \quad \forall k$$

where C_k is the total capacity of machine k during the scheduling period.

(iii) Integer solutions:

The restriction x_i is a non-negative integer must be added.

(b) Production costs other than the cost of trim:

(i) Set up cost:

The objective function must be modified to

$$Z = \sum_{i=1}^n t_i x_i + C \sum_{i=1}^n H(x_i)$$

where $C = \frac{\text{cost of each set up}}{\text{cost of trim loss}}$

$$\text{and } H(x) = \begin{cases} 0 & (\text{if } x = 0) \\ 1 & (\text{if } x \neq 0) \end{cases}$$

(ii) Processing costs incurred other than in the cutting operation:

These, as (a)(ii), are difficult conditions to include.

They require Z to be adjusted to

$$Z = \sum_{i=1}^n t_i x_i + \sum_{i=1}^n b_i x_i$$

where

$$b_i = \frac{\text{cost/unit of extra processing if the } i^{\text{th}} \text{ pattern is used.}}{\text{cost/unit of trim.}}$$

(c) Ways in which the constraints are relaxed:

(i) Underfilling/overfilling orders:

The condition

$$\sum_{i=1}^n a_{ij} x_i = d_j \quad j = 1, 2, \dots, m$$

must be modified to

$$-u_j \leq \sum_{i=1}^n a_{ij} x_i - d_j \leq o_j \quad j = 1, 2, \dots, m$$

where u_j is the amount by which one can underproduce the j^{th} order size.
 and o_j " " " " " " " " overproduce the j^{th} order size.

N.E. The objective function Z is no longer equivalent to Z' when this modification is included.

(ii) Production of off-cuts:

Again the condition

$$\sum_{i=1}^n a_{ij} x_i = d_j \quad j = 1, 2, \dots, m$$

must be modified, this time to

$$\sum_{i=1}^n a_{ij} x_i = d_j + s_j \quad j = 1, 2, \dots, m$$

where s_j ($s_j \geq 0$) is the amount of size w_j produced to stock. If, however, sizes other than those ordered can be treated as off-cuts then the function Z must be modified to

$$Z = \sum_{i=1}^n x_i f(t_i)$$

where

$$f(t) = \text{Min.} \left(\begin{array}{l} \text{cost of scrapping} \\ \text{piece of size } t \end{array} , \begin{array}{l} \text{expected stockholding} \\ \text{cost of piece of size } t \end{array} \right)$$

If material is held in stock as off-cuts it can be re-entered as data for a later problem by treating it as a parent size as in (c)(iii) below.

A slightly different modification to the linear formulation is given by Bernhard (2).

(iii) Several parent sizes:

This enlarges the set of feasible patterns by modifying

$$w = t_i + \sum_{j=1}^m a_{ij} w_j \quad \forall i$$

to

$$w_k = t_i + \sum_{j=1}^m a_{ij} w_j \quad \forall i, \text{ for some } k$$

While it is possible, in the ways outlined above, to introduce many complexities into the linear programming formulation of the trim loss problem, this does not imply that the resulting formulation is of practical significance, as in some cases the linear property has been lost, and in others the size of the enlarged problem makes it computationally infeasible. This will be discussed further in § 2.5 p. 16 and Ch. 3.

2. REVIEW OF CURRENT METHODS

2.1 Introduction

Scheduling problems in which trim loss is an important economic consideration are approached in many different ways. The method used will, in any particular situation, depend on the structure of the firm in which the problem arises as well as the characteristics of the scheduling problem. The methods of solution may be divided into four broad classes which are listed below in order of increasing sophistication:

- (a) manual methods;
- (b) heuristic methods;
- (c) branch and bound techniques;
- (d) linear programming.

Small firms (e.g. steel stock holders) tend to use either manual methods or simple heuristics. There are two main reasons why they do not use the more efficient sophisticated methods: first they do not possess the necessary expertise to employ an advanced method; and secondly the cost of running a complex scheduling method could well outweigh any possible savings on the firm's small turnover. Only in larger companies is it likely that more sophisticated heuristics, branch and bound, or linear programming, will be employed.

2.2 Manual Methods

In this country manual methods are evidently employed more often than the more sophisticated methods. In most firms the actual scheduling is delegated to a low level. The scheduler responsible seems (from interviews with several schedulers) to use a mixture of experience and rule of thumb techniques. The most frequently used method is to schedule the largest sizes demanded first, filling in the rest of each parent size with as many small sizes as possible, the remaining small sizes are then scheduled in the best way the scheduler can see. If this process does not give a satisfactory result (normally the scheduler has an acceptable trim loss percentage which he tries to avoid exceeding) he will try a different arrangement of the large sizes and repeat the procedure. It will normally take two or three attempts before a satisfactory solution is reached. In practice it has been noticed that most schedulers have opportunities to "avoid" excessive trim loss by, for example, sending cut lengths to stock, overfilling orders, or deferring orders to the next time period. These courses of action will necessarily incur

certain costs which, while less easily measured than that of increased trim loss, are still important.

Manual methods are inevitably time-consuming due to the large amount of computation involved in generating alternative schedules. Not untypically, a scheduler can be fully occupied in scheduling just two coil slitting machines.

The process by which the scheduler reaches "a solution" is obviously a satisficing procedure rather than an optimizing one. However, by experience, the scheduler seems to be able to tell when he is reasonably near to the optimum. It has been suggested* that a human scheduler will be reasonably efficient on small problems as he can consider all the feasible patterns, and that he will also be reasonably efficient in situations where there are a very large number of patterns and thus many solutions which are close to the optimum. The worst results are obtained when there are a moderately large number of patterns, but not sufficient to give rise to many solutions near the optimum, and thus where finding the absolute optimum is critical. The suggested efficiency of a human scheduler is as shown in the graph below.

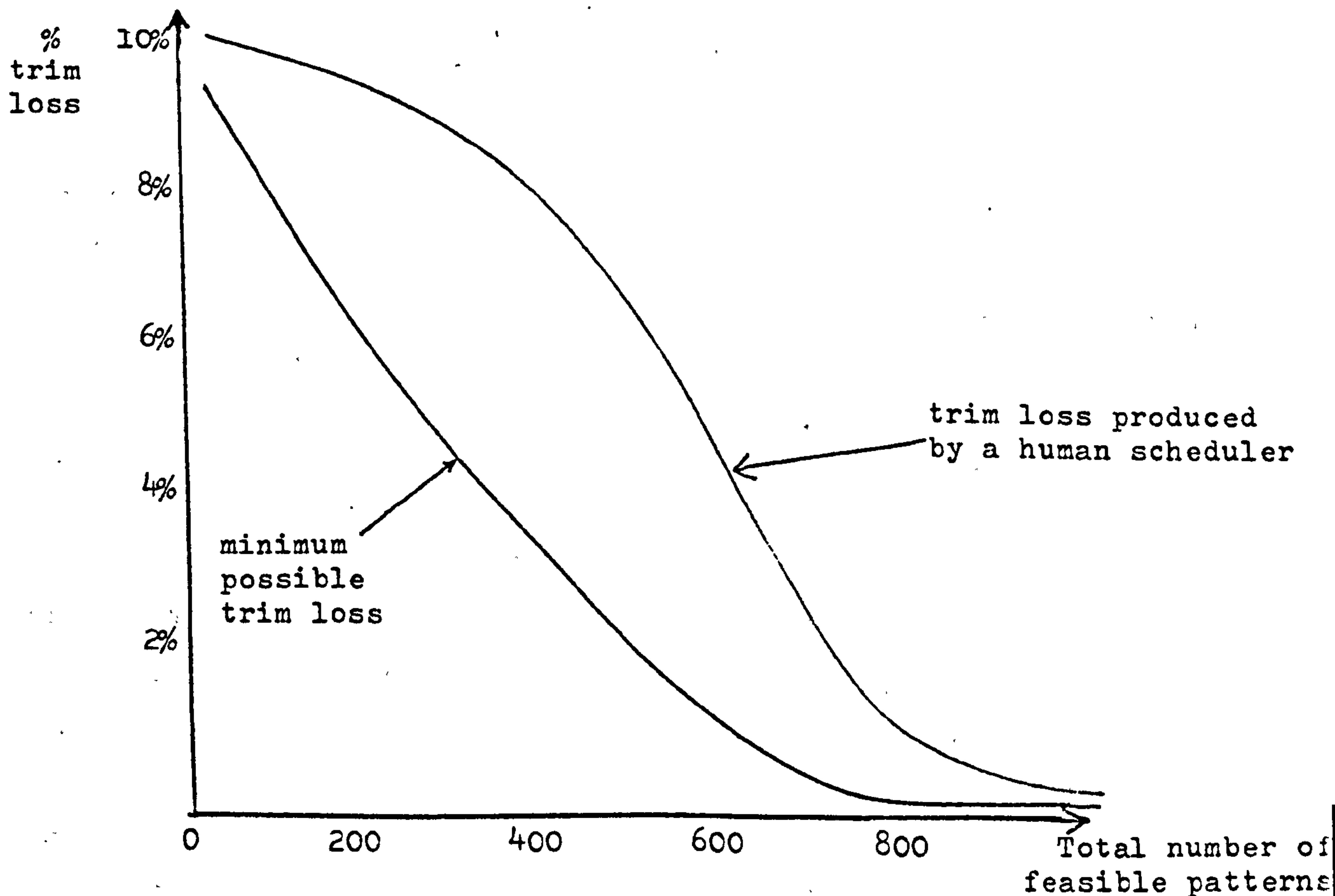


Figure 3 Graph of trim loss and the total number of feasible patterns

* In a private conversation with Mr. A. Hiron of the Paper Industries Research Association.

2.3 Heuristic Methods

Many of the heuristic methods used to solve scheduling problems involving trim loss are formalised computer-based developments of the manual methods outlined above in § 2.2, p. 10. However, some of the more sophisticated heuristics do utilise techniques drawn from the theory of artificial intelligence.

With electronic computation it is possible to check a wider range of feasible patterns (and hence a larger set of solutions) than can any human scheduler, thus better quality solutions can generally be obtained. It is very difficult however to formulate into any computer program the intuitive judgement of a human scheduler in deciding when the search procedure has arrived at a satisfactory solution.

The decision rules employed within the computer programs are built round the structure of the particular set of scheduling problems to be solved. Often the programs are complex, using the particular characteristics of the problems under consideration in order to cut down the number of patterns to be searched before an acceptable solution can be found. As a result of being built around the structure of a particular firm's scheduling problems, computer heuristic methods, while satisfactory for the set of problems for which they were designed, frequently have only a limited range of application.

The decision rules used in most heuristic methods are simply a set of empirical rules which have been found, when used in practice, to yield satisfactory schedules. These heuristic methods often differ only marginally from the manual methods which they have replaced. Marconi's heuristic (15), which is shown below in figure 4, is typical of heuristic methods used in industry which, while a step forward from manual methods, are still relatively unsophisticated.

A different class of heuristic methods is that in which an optimum solution to a simplified form of the scheduling problem is found using one of the more sophisticated techniques (e.g. linear programming). This solution is then modified using a set of empirical decision rules to form a schedule which fits the particular constraints of the scheduling problem under consideration. Normally linear programming is used to obtain the "optimum" solution, e.g. (3), (55) and (56). Caruso and Kokat (3), for example, suggest using linear

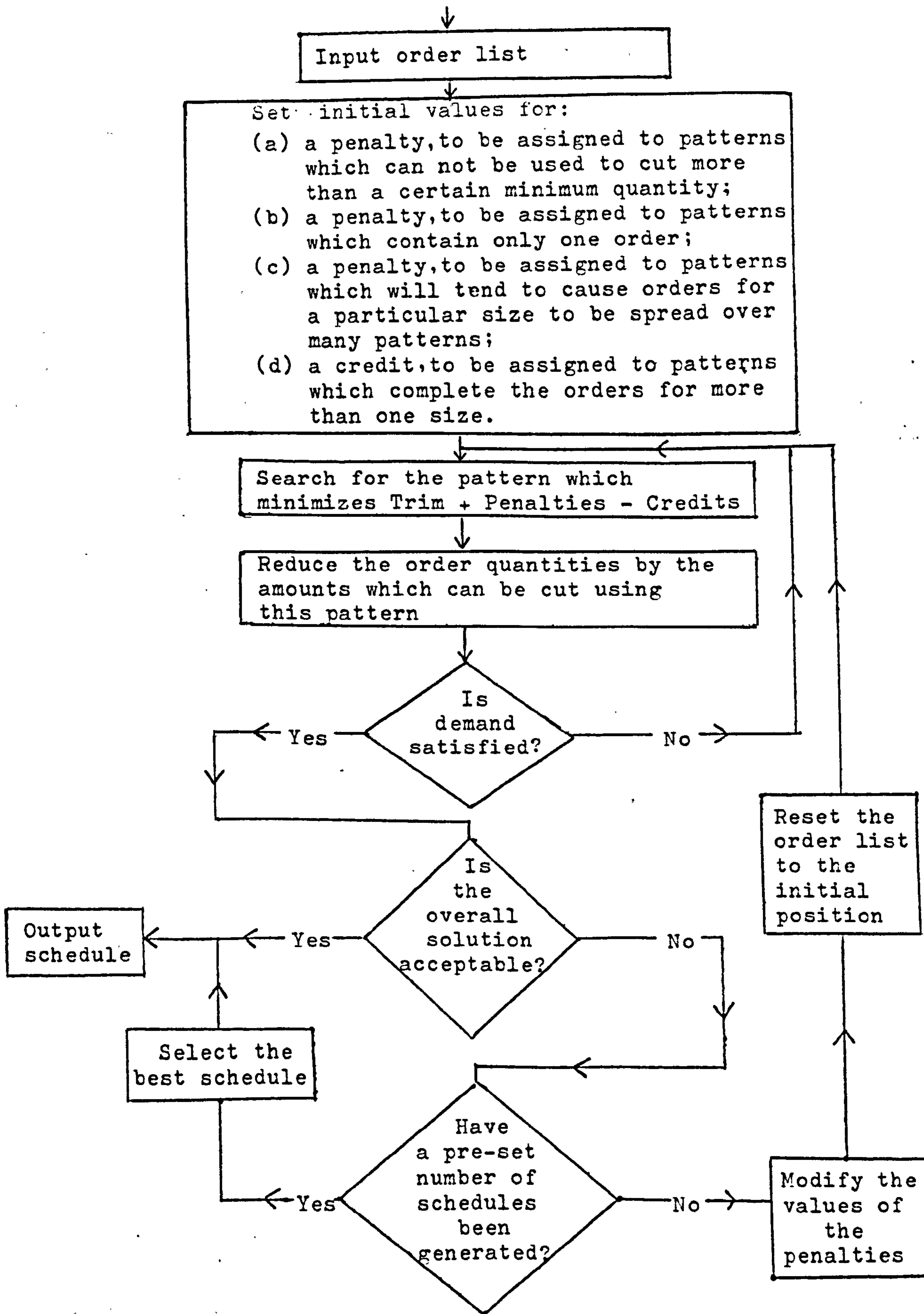


Figure 4. Flow diagram of Marconi's Heuristic.

programming to solve a problem in which an integer solution is required; firstly an optimum non-integer solution is found to the linear programming problem, this is then rounded to give an acceptable, but not necessarily optimum, integer solution. In this, as in many similar methods, branch and bound could equally well have been used in place of linear programming to yield the initial "optimum" solution.

As a result of their intrinsic flexibility, heuristic methods can tackle scheduling problems which can not readily be formulated for solution by a standard sophisticated method. Indeed Hodges (45) argues that this lack of flexibility is the main reason why sophisticated methods are not used widely in the paper industry. Similar criticism of sophisticated methods has arisen in other industries. However, the very flexibility of a heuristic approach can result in a solution method which is so well adapted to the particular problem under consideration that it lacks generality.

Obviously the total improvement which a heuristic method gives over a manual method varies greatly between heuristics. Haessler's (11) claim of a 16% reduction in costs using a heuristic method would, however, seem to be reasonably typical.

This thesis examines one class of heuristic methods which has a degree of generality, pattern enumeration. In a pattern enumeration method feasible patterns are listed (in a predetermined order) until one satisfying certain conditions is found. This pattern is then entered into the solution and the order quantities reduced accordingly. This process is repeated until all the orders are satisfied. The method of Marconi (15), shown in figure 4, uses a simple pattern enumeration approach. The pattern enumeration approach is examined in more detail in chapter 4 and is the basis of the procedures developed and applied in this thesis.

2.4 Branch and Bound Techniques

Branch and bound is the name given to those techniques in which all potential schedules (as distinct from all patterns - a selection of patterns being required to make up a schedule) are considered in an ordered manner. Schedules are eliminated from explicit consideration by:

- (a) feasibility conditions - schedules which will violate one or more of the problem constraints can obviously be excluded from consideration;

(b) dominance conditions - schedules which will be inferior in quality (i.e. incur higher costs) than ones already generated can also be excluded from consideration.

Unlike manual and heuristic methods branch and bound methods will, as a result of searching (implicitly or explicitly) the whole solution set, either terminate at the optimum (given sufficient time) or establish infeasibility.

Pierce (54) shows, in detail, how branch and bound techniques, which of course have a far wider range of application than the trim loss problem, can be used to solve trim problems. In particular, various techniques for cutting down search time by the use of feasibility and dominance constraints are explored by him. The majority of methods using branch and bound for the trim problem are very similar to the approach adopted by Pierce. Wig's method (22) is, however, slightly different in that he utilizes the integer nature of the problem under consideration and is thus able to make use of a theorem of Mathews (49), drawn from number theory, in order to simplify the problem constraints.

While branch and bound techniques are applicable to a wide range of trim problems and, as Pierce points out, are basically simple to apply, there is one large restriction on the range of problems they can tackle. As a result of their discrete structure they can only be applied to integer problems (i.e. to those trim problems described in §1.5(a)(iii) p. 5) or to those which can be approximated to by an integer form. Branch and bound methods can however be easily adapted to encompass several of the more common additional constraints, e.g. a limit on the number of cutting knives (see §1.5(a)(i) p. 5) or the introduction of a set-up cost (see §1.5(b)(i) p.5).

Unfortunately branch and bound techniques do have the drawback of consuming a great deal of computer time when even a moderate sized problem of 20-30 orders is to be tackled. Frequently in order to save computer time it is necessary to terminate the search before it is complete (and hence to generate a solution which is not necessarily optimal). When the search is not completed then the chief advantage of branch and bound over heuristic methods (i.e. optimality) is lost.

2.5 Linear Programming

Linear programming is the most widely used sophisticated optimizing technique for the solution of scheduling problems involving the minimization of trim loss. This wide usage is a result of it being easy to formulate the basic scheduling problem (i.e. demand to be met and trim losses minimized) as a linear programming problem.

The formulation of the basic trim loss problem (see § 1.6 p.6) as:

$$\begin{aligned} \text{minimize } Z &= \sum_{i=1}^n t_i x_i \\ \text{subject to } \sum a_{ij} x_i &= d_j \quad j = 1, 2, \dots, m \end{aligned}$$

is clearly solvable using the simplex method of linear programming. It was first formulated, in 1939, by Kantorovich (14) for solution by linear methods. However since his paper was in Russian using a different notation and method to that of the simplex, it was not until Eisemann's paper (6) in 1957 that the simplex method was used generally in the West for the solution of trim loss problems. In Eisemann's form every possible pattern must be stored in the computer and thus only small problems can be tackled. The number of feasible patterns increases rapidly with the number of sizes ordered and also with the average number of ordered sizes which can be fitted into a parent size. The number of patterns for several examples are shown in Table 1 below (c.f. Bege-Dov's (1) formula in § 1.6 p.6).

Table 1 Total number of feasible patterns

Number of ordered sizes	Average number of ordered sizes per parent size	Number of patterns
3	3	8
2	5	9
5	2	15
5	3	88
7	4	379
9	3	381
14	3	1295
22	2	1395
23	2	2060
6	11	7960
22	4	Greater than 10,000

As can be seen the number of patterns becomes prohibitive on even a medium sized problem with 22 demand sizes, and in many of the problems

found in practice, the number of patterns can exceed a million. This makes the method impractical even on a large computer.

In 1961 Gilmore and Gomory (9) suggested a way of overcoming this difficulty by using the revised simplex method developed by Dantzig in which only the basic square matrix needs to be stored. In Gilmore and Gomory's method, patterns are only generated (by a dynamic programming algorithm) as they are required to be added to the basis in order to improve the solution. Although this modification reduced storage requirements to manageable proportions, the run times were still quite large; Gilmore and Gomory (10) gave an example with 30 ordered sizes and an average of 5 ordered sizes per parent size which took 13 mins. to run on an IBM 7090 (or 1357 mins. on an IBM 1620). To reduce this run time they suggested an improved method (10) in which the dynamic programming section for selecting new patterns is replaced by a search procedure. This reduced run times by a factor of 5. Their method is used as a basis for the standard trim loss packages produced by both IBM and ICL.

Since this basic linear programming method was published there have been several papers (e.g. (2), (3), (16), (50), (60)) showing how a linear programming method can be used to solve particular specialized trim loss problems. However, the basic theory for the solution of one dimensional trim loss problems by linear programming has not advanced past that of Gilmore and Gomory, and it must be stressed that their method is only relevant to certain basic forms of the scheduling problem. Other forms of the scheduling problem more likely to be met in practice involve non-linearities which linear programming can not tackle. These more realistic problems will be considered in the next chapter.

3. LIMITATIONS OF CURRENT METHODS

3.1 Introduction

The scheduling methods described in the previous chapter differ in many respects. Generally, as has been pointed out, the more sophisticated the method, the more limited is the objective and the range of problems which can be solved. In practice relatively crude manual methods predominate.

The techniques which are available in computer package form (normally based on a linear programming approach) are not suitable for many of the smaller firms facing a trim loss problem because they do not possess the expertise to understand the packages and therefore to use them effectively.

Both linear programming and branch and bound methods consume a great deal of computer time in processing a realistic problem. Thus in industries with small amounts to be scheduled or with inexpensive products (e.g. the paper slitting problem of chapter 5) the cost in computer time of obtaining an optimum will not be covered by the savings produced over a non-optimal manual solution.

Moreover, the reason why sophisticated methods can not be applied is that the additional economic considerations and technical constraints needed to model any practical problem cannot easily be accommodated within the formulations used in standard packages. Several of the main difficulties (set-up costs, the integer problem, under and over production) are outlined in the later sections of this chapter together with the different ways which have been used to overcome them. Generally it is more difficult to include additional conditions within a linear programming method than within a branch and bound method. Even when additional conditions can be included within a linear programming or branch and bound formulation, they complicate the formulation considerably and are likely to give rise to a significant increase in solution time. Heuristic methods, on the other hand, being developed for individual problems can handle these additional conditions, but as a result are very limited in their range of application.

3.2 Throughput Constraints and Set-up Costs

Several attempts have been made to develop procedures which take account of limited capacity of the cutter or the parallel difficulty of set-up times. Some methods are an amalgam of linear programming

and heuristic methods [(3), (55), (56)] while others are completely heuristic [(11), (12), (15)].

Since the schedules produced by linear programming methods have the property that when demands must be exactly satisfied, the number of set-ups required by the schedules will equal the number of demand sizes, it is necessary to use some technique - normally involving heuristics - to reduce the number of set-ups to an acceptable number. Generally this requires the demand constraints to be weakened. One way in which this can be achieved is by specifying an acceptable range of demand for some or all of the demand sizes. Caruso and Kokat (3), for example, obtain a linear programming solution which only minimizes trim and then use inter-active programming to both reduce the number of set-ups required and obtain an integer solution. Poirier (55) has developed a similar approach but one in which heuristics, rather than human intervention, are used to modify the initial solution. Potts (56) uses linear programming to obtain a solution, but only after heuristics have been used to relax some demand constraints so that sizes may be grouped according to demand. He thereby reduces the number of constraints and hence the number of set-ups required by any solution. It must be noted that whenever any of these modifications of linear programming are used, then its main advantage, that of optimality, is lost. It is interesting that a problem in the financial field (61) with a similar mathematical form has proved equally difficult to solve using linear programming methods alone. Theoretically it is possible to formulate this problem in integer programming terms but the cost of solving such a formulation is prohibitive.

Methods which are entirely heuristic cannot easily be classified since they are usually developed to suit particular problem characteristics. An interesting common feature of the methods developed by Haessler [(11) and (12)] and Marconi (15) is that more than one solution is obtained by changing parameters within the heuristics. The best schedule is then selected from the set of solutions obtained. Haessler describes this process as a "multiple pass procedure". This sort of experimentation is frequently necessary with heuristic methods as, whilst the method may significantly reduce the average number of set-ups required over a range of problems, for any particular problem it is difficult to predict at the outset whether the solution obtained by a particular set of parameter values will be entirely satisfactory.

Ideally one would like to be able to rank the different methods by their performance on a standard set of test problems. Pegels (52) does this for three methods which consider similar problems in the construction of schedules for corrugated cardboard. Unfortunately it is not possible to do this over a wider range of problems as each method has only a narrow range of application.

3.3 Integer Problems

The problems in which it is necessary to include integer variables often also contain through-put constraints or set-up costs. This is no coincidence, as the integer restriction is often imposed as a way of improving throughput or reducing set-up costs. For example, in roll slitting problems it is frequently stipulated that no machine shall be reset part way through a roll but that all knife changes shall occur when a parent roll is to be changed. As a result of this close relationship, the methods for handling these integer problems [(3), (13) (22), (50) and (56)] are very similar, if not identical, to those for handling the problem of set-up costs.

One paper, however, does outline a method which is significantly different. Instead of regarding the integer constraint as an extra restriction to be added to a standard method, Wig (22) considers the integer restriction as the fundamental property of the solution to be sought. He can thus use the properties of integers from the Theory of Numbers to develop a solution. A theorem of Mathews (49) is used to cut down all the demand constraints to one equation which can then be solved by search methods. This gives a very simple method in situations where the number of rolls to be cut is small and where the order quantities for each size are also small. However, as the magnitude of the quantities involved increases, so does the complexity of the single equation produced and hence the search time.

Theoretically, as with problems involving set-up costs, one could use integer programming methods but they are not widely used in practice for two reasons. First, integer programming formulations of practical trim problems take a long time to solve even on a fast computer; they are therefore very expensive. Secondly, practical integer trim problems frequently contain additional factors which can not be included within an integer programming formulation. Thus the need to include integer constraints, as with the need to include through-put constraints or set-up costs, has further inhibited the development of methods which are generally applicable.

3.4 Under and over Production

The relaxation of the demand constraints so that under or over production of the demand sizes by a fixed percentage (normally 5 - 10%) is allowed, or even so that extra stock sizes can be produced, can easily be included within a linear programming formulation of the simple trim problem as shown in §1.6 p.6 or slightly differently as in Bernhard (2). However, if the scheduling problem were ever one of simply minimizing trim loss and required no consideration of set-up time, integer solutions, or other complicating factors, then it would normally be possible to obtain acceptable solutions by linear programming without recourse to under or over production. The opportunity to under or over produce tends to be only allowed in situations where complications do exist and where it is known that acceptable trim losses and processing costs can only be obtained if there is a relaxation of the demand constraints. The relaxation of the demand constraints is therefore best considered as it arises in practice [(3), (13), (22) and (56)] viz. as a means, perhaps costly, of obtaining feasible and satisfactory solutions when there are complications of the type described in §3.2 and §3.3. Each method handles under and over production differently but in none is it considered to be the main feature of the problem under consideration. Thus the possibility of under or over production in a scheduling situation merely adds a further degree of complexity to what is already a rather unmanageable problem.

3.5 Summary

Manual methods are subjective and relatively crude, yet they have the advantage of being flexible, allowing the scheduler to take account, to the best of his ability, of all the economic considerations and technical constraints of the problem. Despite their simplicity, manual scheduling methods are by no means cheap as a great deal of time-consuming arithmetic is required to develop a schedule.

Published heuristic methods are usually developed for particular problems (which they model very well) and consequently they have a limited range of application. They are more structured than manual methods, and are able to cope with most of the economic considerations and technical constraints encountered in practice.

Theoretically any trim loss problem could be formulated for solution using branch and bound. In practice, however, branch and bound methods are not widely used as, faced with the large numbers of possible

patterns in practical problems, and hence the even larger number of feasible solutions, the formulation of an efficient search algorithm for all but the simplest of problems is very difficult.

Linear programming and associated mathematical programming methods, in particular that of Gilmore and Gomory (10), may be used to obtain solutions to simple problems. When more complex, and more typical, problems are met, a mathematical programming approach runs into difficulties. Even when a practical problem can be formulated for solution by mathematical programming it is frequently, as with branch and bound, very expensive in computer time and storage.

4. THE PATTERN ENUMERATION METHOD

4.1 Introduction

None of the techniques described in the literature for the solution of the trim problem have been applied to a wide range of practical problems. The object of this study is to develop a basic approach to the trim problem which, with relatively simple adaptations, could be used for a wide range of practical problems having differing constraints and cost considerations. The method, while not necessarily yielding optimum solutions, must yield schedules which are on the whole significantly better than those generated manually, and which are more realistic than those obtained using "optimization" methods which necessarily require an oversimplification of the total scheduling problem. Having surveyed the range of problems faced in scheduling cutting operations it is clear that any method which is designed to be used in a wide range of practical problems should meet the following criteria:

(a) It should be possible to include within the formulation of the method not only the constraint that demand be satisfied, but also the many other common constraints which arise in practice, e.g. limitations on the number of set-ups, the need for integer solutions, limitation on the number of cutting edges;

(b) It should be possible to consider, either implicitly or explicitly, not only the cost of trim but also other relevant production costs, e.g. reprocessing costs, set-up costs, storage costs;

(c) It should be possible to include within the method the many currently used techniques for reducing excessive trim, e.g. over and under production, production to stock, stocking of off-cuts;

(d) The method should not require the large amounts of computer time and store required by classical methods. Such large scale use of computing facilities is expensive and the cost can, in many situations, be of equivalent magnitude to that saved in efficient production scheduling;

(e) The method should be relatively easily understood and applied. Despite the economic significance of the trim problem, its solution is delegated to a low level in many organizations, the scheduler frequently not having the training needed to apply sophisticated mathematical techniques.

4.2 The Basic Pattern Enumeration Method

Whatever the character of particular trim problems (in terms of costs and constraints) the problem which the scheduler has to solve is to choose a few patterns from the vast number which can be made up from the sizes to be scheduled (see Table 1 §2.5 p.16). In the pattern enumeration method described below the problem is reduced to manageable proportions in the following ways:

- (a) A sequential procedure is followed. Once a pattern has been chosen the number of rolls or bars to be cut by it will be decided. The order quantities will then be reduced by the amounts "cut" by the chosen pattern. These reduced order quantities will then constitute a reduced problem which will be solved in the same way until all the demand has been scheduled;
- (b) There is no attempt to examine all possible patterns. Once a pattern meeting certain acceptance criteria is found this will be introduced into the solution.

Clearly the crucial part of this type of solution procedure is the method used to select a pattern. Consider the set of all possible patterns which cut sizes w_1, w_2, \dots, w_m from a parent size W , namely $\{\underline{a}_1, \underline{a}_2, \dots, \underline{a}_n\}$ where $(\underline{a}_i = a_{i1}, a_{i2}, \dots, a_{im})$ (as in §1.6 p.6).

Then for \underline{a}_i to be a feasible pattern:-

$$\sum_{j=1}^m a_{ij} w_j \leq W$$

Consider a lexicographic ordering of the patterns $\underline{a}_1, \underline{a}_2, \dots, \underline{a}_n$ (c.f. Gilmore and Gomory (9)) such that for any two patterns in the list \underline{a}_{i_1} and \underline{a}_{i_2}

$$\text{then } i_1 < i_2 \Leftrightarrow \exists j \text{ s.t. } 1 \leq j \leq m$$

$$a_{i_1 j} > a_{i_2 j}$$

$$a_{i_1 k} = a_{i_2 k} \quad \forall k < j$$

This ordering is unique.

The ordering is illustrated in the example below where

$$\begin{array}{lll}
 m = 3 & W = 13 & \\
 w_1 = 5 & w_2 = 4 & w_3 = 3
 \end{array}$$

			Trim
then	\underline{a}_1	= (2, 0, 1)	---- 0
	\underline{a}_2	= (2, 0, 0)	---- 3
	\underline{a}_3	= (1, 2, 0)	---- 0
	\underline{a}_4	= (1, 1, 1)	---- 1
	\underline{a}_5	= (1, 1, 0)	---- 4
	\underline{a}_6	= (1, 0, 2)	---- 2
	\underline{a}_7	= (1, 0, 1)	---- 5
	\underline{a}_8	= (1, 0, 0)	---- 8
	\underline{a}_9	= (0, 3, 0)	---- 1
	\underline{a}_{10}	= (0, 2, 1)	---- 2
	\underline{a}_{11}	= (0, 2, 0)	---- 5
	\underline{a}_{12}	= (0, 1, 3)	---- 0
	\underline{a}_{13}	= (0, 1, 2)	---- 3
	\underline{a}_{14}	= (0, 1, 1)	---- 6
	\underline{a}_{15}	= (0, 1, 0)	---- 9
	\underline{a}_{16}	= (0, 0, 4)	---- 1
	\underline{a}_{17}	= (0, 0, 3)	---- 4
	\underline{a}_{18}	= (0, 0, 2)	---- 7
	\underline{a}_{19}	= (0, 0, 1)	---- 10
	\underline{a}_{20}	= (0, 0, 0)	---- 13

The patterns can be split into two groups, those which are dominated and those which are non-dominated. A pattern \underline{a}_{i_1} is defined to be dominated if

$$\exists i_2 \neq i_1 \text{ s.t. } a_{i_2 j} \geq a_{i_1 j} \quad \forall j$$

e.g. \underline{a}_2 above is dominated by \underline{a}_1 . Those patterns which are dominated by another pattern can be eliminated from consideration, as at each stage of the solution procedure an attempt will be made to use as much of the parent roll as possible (a dominated pattern must, from its definition, have a trim such that at least one ordered size may be fitted into it). In the above example this will reduce the patterns for consideration to:-

	Trim
(2, 0, 1)	---- 0
(1, 2, 0)	---- 0
(1, 1, 1)	---- 1



	Trim
(1, 0, 2) ---	2
(0, 3, 0) ---	1
(0, 2, 1) ---	2
(0, 1, 3) ---	0
(0, 0, 4) ---	1

The set of non-dominated patterns for any list of demand sizes can be generated by the method shown in the flow diagram (Fig. 5) below. A Fortran program segment to generate these patterns is shown as Program 1, Appendix 1.2, p. 111.

Using this enumeration of patterns as the basic step, a method for solving simple trim problems can now be outlined:

- (a) Sort the demand list into descending order of size;
- (b) Set an acceptable trim. In situations where manual scheduling is employed, the scheduler will have some acceptable trim figure which he aims to keep below on each pattern used. This figure can be used as a first estimate of acceptable trim. However, improved overall solutions can be obtained by tuning this parameter on several sets of test data, as is demonstrated in the case studies following;
- (c) Enumerate the non-dominated patterns until one having an acceptable trim is found. If all patterns are enumerated and none have an acceptable trim then choose that with the least trim;
- (d) Calculate the maximum amount which can be cut with this pattern without overfilling the order, and reduce the demand quantities accordingly. (In bar cropping problems the amount to be cut by a pattern will be the number of bars to be cut which will necessarily be integer, whereas for roll slitting problems the amount to be cut will represent the length of unrolled coil to be slit by this pattern and therefore need not necessarily be integer);
- (e) If there is still unsatisfied demand return to (c). If all the demand has been satisfied the solution procedure is complete.

The method outlined above will form the basis for those methods developed in this thesis. Pattern enumeration methods similar to this have been used by Haessler (11), Johns (13), Marconi (15) and Pierce (16) but whereas they concentrated on particular problems,

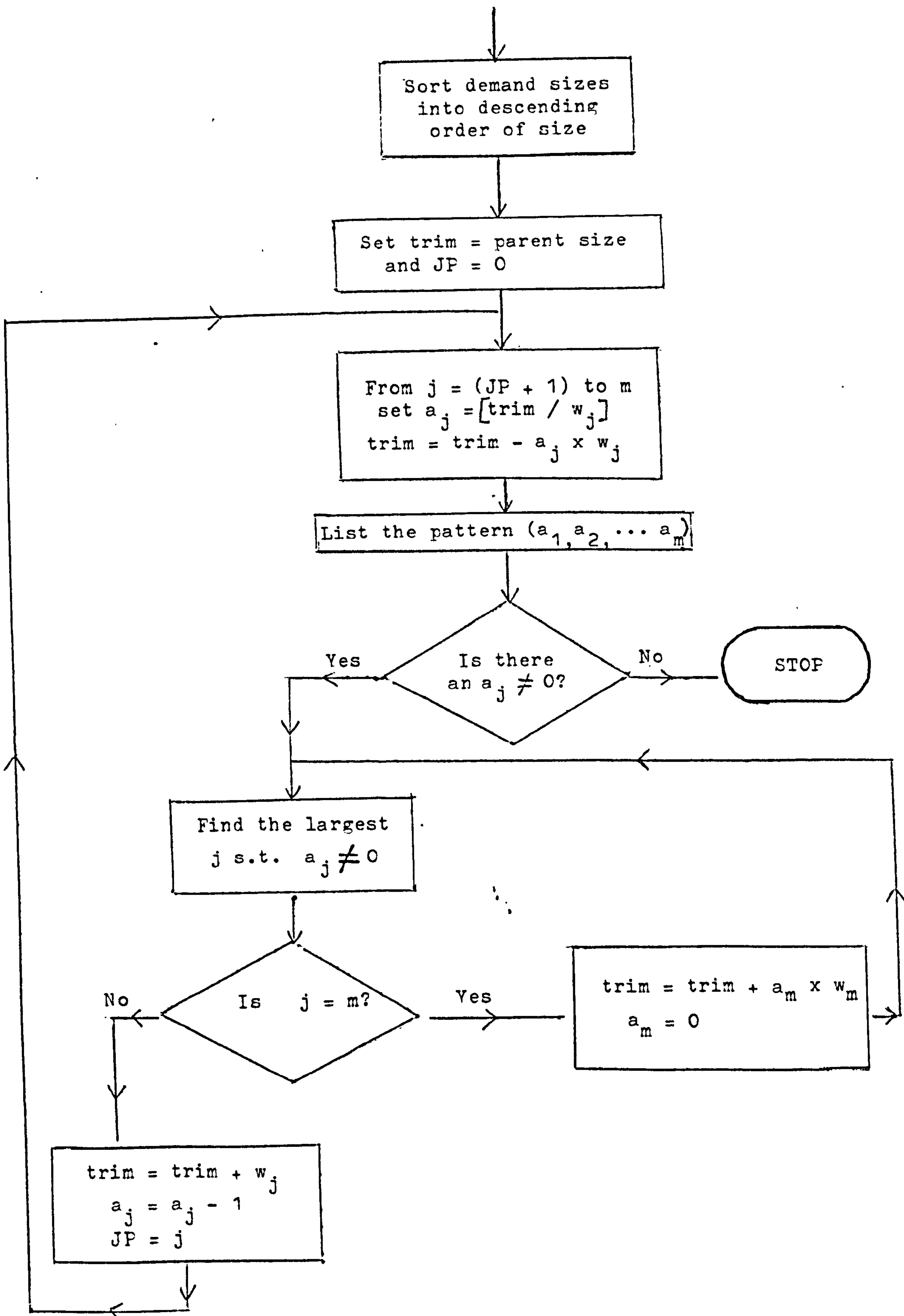


Figure 5. Flow diagram of the method used to enumerate all non-dominated patterns.

here the aim is to develop a general approach.

4.3 A Numerical Example

The following example is intended to demonstrate the mechanics of the pattern enumeration method described above. In the methods developed later for the three case studies, the basic method is adapted and embellished to take account of particular constraints and objectives. Some of the more useful adaptations are discussed in §4.4.

It is assumed here, however, that there are no constraints on the number of set-ups, number of sizes per pattern, nor is an integer solution required. The objective is assumed to be to produce a schedule with a reasonably small trim loss. It is further assumed that demands must be met exactly. In real problems it is most unlikely that these simplifying assumptions would apply.

Data set 1: Parent size 12 000

Order list:	Size	Demand
	939	8
	1200	32
	1000	51
	1225	31
	530	20
	1450	164
	5900	10
	300	36
	5075	38
	725	4
	2425	4
	2100	16
	3300	28
	4000	214

Step (a) Sort the demand list into order viz:-

5900, 5075, 4000, 3300, 2425, 2100, 1450, 1225, 1200, 1000, 939, 725, 530, 300.

Step (b) Set an acceptable trim; for this example 10 or less will be considered to be an acceptable trim.

Iteration 1

Step (c) Enumerate patterns until one with acceptable trim is found.

Size:

5900 5075 4000 3300 2425 2100 1450 1225 1200 1000 939 725 530 300

Quantity required:

	10	38	214	28	4	16	164	31	32	51	8	4	20	36	Trim	
$\underline{a}_1 =$	(2,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0)	200
$\underline{a}_2 =$	(1,	1,	0,	0,	0,	0,	0,	0,	0,	1,	0,	0,	0,	0,	0)	25
$\underline{a}_3 =$	(1,	1,	0,	0,	0,	0,	0,	0,	0,	0,	1,	0,	0,	0,	0)	86
$\underline{a}_4 =$	(1,	1,	0,	0,	0,	0,	0,	0,	0,	0,	0,	1,	0,	1)	0	

Step (d) Calculate the maximum amount which can be cut by the pattern now.

10÷1 units can be cut before the demand for size 5900 is satisfied.

38÷1 " " " " " " " " " " 5075 " "

4÷1 " " " " " " " " " " 725 " "

36÷1 " " " " " " " " " " 300 " "

Thus the maximum number of units which can be cut is 4. This reduces the quantities remaining of sizes 5900, 5075, and 300, to 6, 34, and 32 respectively, while the demand for size 725 is completely satisfied and it can be deleted from the demand list.

Step (e) Return to (c)

Iteration 2

Step (c)

Size: 5900 5075 4000 3300 2425 2100 1450 1225 1200 1000 939 530 300

Quantity remaining:

	6	34	214	28	4	16	164	31	32	51	8	20	32	Trim	
$\underline{a}_1 =$	(2,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0)	200
$\underline{a}_2 =$	(1,	1,	0,	0,	0,	0,	0,	0,	0,	1,	0,	0,	0,	0)	25
$\underline{a}_3 =$	(1,	1,	0,	0,	0,	0,	0,	0,	0,	0,	1,	0,	0,	0)	86
$\underline{a}_4 =$	(1,	1,	0,	0,	0,	0,	0,	0,	0,	0,	0,	1,	1,	1)	195
$\underline{a}_5 =$	(1,	1,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	3,	3)	125
$\underline{a}_6 =$	(1,	0,	1,	0,	0,	1,	0,	0,	0,	0,	0,	0,	0,	0)	0

Step (d)

Maximum number of units which can be cut by the pattern =

Minimum (6/1, 214/1, 16/1) = 6.

Step (e) Return to (c)

Iteration 3

Step (c)

Size: 5075 4000 3300 2425 2100 1450 1225 1200 1000 939 530 300

Quantity remaining:

	34	208	28	4	10	164	31	32	51	8	20	32	Trim
a_1	(2,	0,	0,	0,	0,	1,	0,	0,	0,	0,	0,	1)	100
a_2	(2,	0,	0,	0,	0,	0,	1,	0,	0,	0,	1,	0)	95
a_3	(2,	0,	0,	0,	0,	0,	1,	0,	0,	0,	0,	2)	25
a_4	(2,	0,	0,	0,	0,	0,	0,	1,	0,	0,	1,	0)	120
a_{20}	(1,	1,	0,	0,	0,	1,	0,	0,	1,	0,	0,	1)	175
a_{21}	(1,	1,	0,	0,	0,	1,	0,	0,	0,	1,	1,	0)	6

Step (d)

Number of units to be cut = Minimum (34/1, 208/1, 164/1, 8/1, 20/1) = 8.

Step (e) Return to (c)

Iteration 4

Step (c)

Size: 5075 4000 3300 2425 2100 1450 1225 1200 1000 530 300

Quantity remaining:

	26	200	28	4	10	156	31	32	51	12	32	Trim	
a_1	(2,	0,	0,	0,	0,	1,	0,	0,	0,	0,	1,)	100
a_{50}	(1,	0,	1,	1,	0,	0,	0,	1,	0,	0,	0,)	0

Step (d)

Number of units to be cut = Minimum (26/1, 28/1, 4/1, 32/1) = 4

Step (e) Return to (c)

Iteration 5

Step (c)

Size: 5075 4000 3300 2100 1450 1225 1200 1000 530 300

Quantity remaining:

	22	200	24	10	156	31	28	51	12	32	Trim
a_1	(2,	0,	0,	0,	1,	0,	0,	0,	0,	1)	100
a_{50}	(1,	0,	1,	1,	0,	1,	0,	0,	0,	1)	0

Step (d)

Number of units to be cut = Minimum (22/1, 24/1, 10/1, 31/1, 32/1) = 10.

Step (e) Return to (c)

Iteration 6

Step (c)

Size: 5075 4000 3300 1450 1225 1200 1000 530 300

Quantity remaining:

	12	200	14	156	21	28	51	12	22				
\underline{a}_1	=	(2,	0,	0,	1,	0,	0,	0,	1) Trim	100	
\underline{a}_{66}	=	(1,	0,	1,	0,	1,	2,	0,	0;	0) Trim	0

Step (d)

Number of units to be cut = Minimum (12/1, 14/1, 21/1, 28/2) = 12.

Step (e) Return to (c)

Iteration 7

Step (c)

Size: 4000 3300 1450 1225 1200 1000 530 300

Quantity remaining:

	200	2	156	9	4	51	12	22		
\underline{a}_1	=	(3,	0,	0,	0,	0,	0) Trim	0

Step (d)

Number of units to be cut = 200/3 = 66.67

Step (e) Return to (c)

Iteration 8

Step (c)

Size: 3300 1450 1225 1200 1000 530 300

Quantity remaining:

	2	156	9	4	51	12	22				
\underline{a}_1	=	(3,	1,	0,	0,	0,	1,	0,) Trim	120
\underline{a}_2	=	(3,	1,	0,	0,	0,	0,	2) Trim	50
\underline{a}_3	=	(3,	0,	1,	0,	0,	1,	1) Trim	45
\underline{a}_4	=	(3,	0,	1,	0,	0,	0,	2) Trim	275
\underline{a}_5	=	(3,	0,	0,	1,	0,	1,	1) Trim	70
\underline{a}_6	=	(3,	0,	0,	1,	0,	0,	3) Trim	0

Step (d)

Number of units to be cut = Minimum (2/3, 4/1, 22/3) = .67

Step (e) Return to (c)

Iteration 9

Step (c)

Size: 1450 1225 1200 1000 530 300

Quantity remaining:

	156	9	3.33	51	12	20	Trim
\underline{a}_1	(8,	0,	0,	0,	0,	1)	100
\underline{a}_{25}	(6,	0,	2,	0,	0,	3)	0

Step (d)

Number of units to be cut = Minimum (156/6, 3.33/2, 20/3) = 1.67

Step (e) Return to (c)

Iteration 10

Step (c)

Size: 1450 1225 1000 530 300

Quantity remaining:

	146	9	51	12	15	Trim
\underline{a}_1	(8,	0,	0,	0,	1)	100
\underline{a}_{20}	(6,	0,	3,	0,	1)	0

Step (d)

Number of units to be cut = Minimum (146/6, 51/3, 15/1) = 15

Step (e) Return to (c)

Iteration 11

Step (c)

Size: 1450 1225 1000 530

Quantity remaining:

	56	9	6	12	Trim
\underline{a}_1	(8,	0,	0,	0)	400
\underline{a}_{74}	(3,	0,	5,	5)	0

Step (d)

Number of units to be cut = Minimum (56/3, 6/5, 12/5) = 1.2

Step (e) Return to (c)

Iteration 12

Step (c)

Size: 1450 1225 530

Quantity remaining:

	52.4	9	6	
\underline{a}_1	(8	0,	0)	Trim 400
\underline{a}_{48}	(0,	2,	18)	10

Step (d)

Number of units to be cut = Minimum (9/2, 6/18) = .33

Step (e) Return to (c)

Iteration 13

Step (c)

Size: 1450 1225

Quantity remaining:

	52.4	8.33	
\underline{a}_1	(8,	0)	Trim 400
\underline{a}_2	(7,	1)	625
\underline{a}_3	(6,	2)	850
\underline{a}_4	(5,	3)	1075
\underline{a}_5	(4,	5)	75
\underline{a}_6	(3,	6)	300
\underline{a}_7	(2,	7)	525
\underline{a}_8	(1,	8)	750
\underline{a}_9	(0,	9)	975

In this case enumeration is completed without finding a pattern with acceptable trim so the one with lowest trim is chosen, i.e. \underline{a}_5 .

Step (d)

Number of units to be cut = Minimum (52.4/4, 8.33/5) = 1.67

Step (e) Return to (c)

Iteration 14

Step (c)

Size: 1450

Quantity remaining:

	45.74	
\underline{a}_1	(8) 400

Step (d)

Number of units to be cut = 45.74/8 = 5.74

Step (e) The demand is now completely satisfied and so the solution procedure is complete.

The complete solution is shown below.

Pattern 1

To be used to cut 4 units

1 of 5900

1 of 5075

1 of 725

1 of 300

Trim = 0

Pattern 2

To be used to cut 6 units

1 of 5900

1 of 4000

1 of 2100

Trim = 0

Pattern 3

To be used to cut 8 units

1 of 5075

1 of 4000

1 of 1450

1 of 939

1 of 530

Trim = 6

Pattern 4

To be used to cut 4 units

1 of 5075

1 of 3300

1 of 2425

1 of 1200

Trim = 0

Pattern 5

To be used to cut 10 units

1 of 5075

1 of 3300

1 of 2100

1 of 1225

Trim = 0

Pattern 6

To be used to cut 12 units

1 of 5075

1 of 3300

1 of 1225

2 of 1200

Trim = 0

Pattern 7

To be used to cut 66.67 units

3 of 4000

Trim = 0

Pattern 8

To be used to cut .67 units

3 of 3300

1 of 1200

3 of 300

Trim = 0

Pattern 9

To be used to cut 1.67 units

6 of 1450

2 of 1200

3 of 300

Trim = 0

Pattern 10

To be used to cut 15 units

6 of 1450

3 of 1000

1 of 300

Trim = 0

Pattern 11

To be used to cut 1.2 units

3 of 1450

5 of 1000

5 of 530

Trim = 0

Pattern 12

To be used to cut 0.33 units

2 of 1225

18 of 530

Trim = 10

Pattern 13

To be used to cut 1.67 units

4 of 1450

5 of 1225

Trim = 75

Pattern 14

To be used to cut 5.74 units

8 of 1450

Trim = 400

There are over 1 000 000 feasible patterns for this problem. The procedure above generated only 381. Of these 381, many were repetitions and could have been avoided by using the modifications described in §4.5.

The trim generated by the first twelve patterns was only 0.003%. The thirteenth pattern increased this to 0.01% and the last to 0.15%. This increase in trim towards the end of the scheduling procedure is typical of the schedules generated both by pattern enumeration and by other heuristics. The success of a method will depend on the extent to which intractable residual problems can be avoided.

The number of patterns used is fourteen, the same as that had a linear programming method been used. In many situations this result would be infeasible or unsatisfactory due to constraints or costs associated with set-ups. An enumeration procedure such as this is easily adapted in this respect, as is shown in the case studies. Corresponding modifications to classical mathematical programming techniques are, at best, difficult.

Likely adaptations of the procedure for application to real problems are discussed in the next section, whilst ways in which the procedure can be made more efficient are considered in §4.5.

4.4 Pattern Enumeration Techniques

The simple method outlined in the previous section is obviously limited in terms of both its solution quality and its range of application. The case studies of chapters 5, 6 and 7 illustrate an approach by which these limitations can be overcome. This section considers some of the techniques which can be used to give pattern enumeration the flexibility needed by the approach.

The character of the solutions obtained by a pattern enumeration scheduling method depend crucially on the nature of its enumeration phase. Three main factors determine which pattern is chosen to be entered into solution namely:

- the test used to establish whether a pattern is acceptable;
- the order in which patterns are enumerated;
- limitations placed on the patterns to be enumerated.

There are a variety of techniques by which these factors can be manipulated:^{*}

(a) by varying the sequence in which demand sizes appear in the order list. The descending size sequence used in § 4.2 & § 4.3 is convenient because it generates only non-dominated patterns. Other sequences will generate not only all the non-dominated patterns but also some of the dominated ones as well. Nevertheless they may have counter-balancing advantages. Demand sizes appearing high in the sequence will appear in the earlier patterns. They are, thus, more likely to be included in the chosen pattern than those at the foot of the list of demand sizes.

Previous work (13) and (16) on pattern enumeration methods has assumed that there is a "best" sequence of demand sizes. Johns (13) suggests that this is in descending order of the trim left if each order size had to be scheduled only in combination with itself, while Pierce (16) uses the sequence of descending size outlined above. The suggestion developed in this thesis, however, is that there is no "best" sequence but

* The comments on the effectiveness of these techniques are based on the experience gained in the three case studies, not merely on the example (Data set 1, Appendix 1.1, p.111) used to illustrate the point here.

that one should be chosen in such a way as to tailor the patterns selected at each stage so that by avoiding intractable residual problems, an efficient overall solution is produced.

There are many different sequences which could be used but the four main alternatives are those of descending or ascending size or demand:

(i) Descending size. This will tend to force larger sizes into solution. It is therefore useful in situations where it would be impossible to find an acceptable pattern if only large sizes were left towards the end of the solution procedure and where it is thus necessary to force large sizes into solution while there are still sufficient small sizes for them to be matched with. In the first pattern enumerated from the test data with a zero trim, the presence of longer lengths can clearly be seen:

1 of 5900

1 of 5075

1 of 725

1 of 300

This ordering has the additional advantage of being generally quicker than many of the others as only non-dominated patterns are generated when the method shown in Fig. 5, p.27 is used;

(ii) Ascending size. This ordering has essentially the opposite effect of ordering in descending size. It has limited usefulness as small sizes, which are often useful in reducing trim when few sizes remain, are used up early in the procedure. There is one situation where it is of use; when there are a limited number of cutting edges, one would wish to use up the small sizes whenever possible as these would produce high trim if they had to be cut together at the end of the solution procedure. The first pattern selected with this ordering was:

40 of 300

(iii) Descending demand. This will tend to produce patterns which can be used for long production runs. In situations where there are tolerances on demand

quantities this can help reduce the number of set-ups (provided care is taken with the small quantities left over, see the case study of Ch. 5). On the test data this selected as the first pattern:

3 of 4000

which can be used for $71\frac{1}{3}$ units, whereas that produced by (i) could only be used for 4 units, and that of (ii) for .9 unit. This ordering, while not generating individual patterns quickly, tends to be reasonably fast overall as fewer patterns have to be generated;

(iv) Ascending demand. Such an ordering could be useful where it was necessary to handle the small quantities first, e.g. when demand must be satisfied to within 5% and an integer solution is required, then a difference of one unfilled unit in an order for four is far more critical than one in an order for 40. The first pattern selected with this ordering was:

12 of 725

1 of 2100

1 of 1200

which could have been used to cut only $\frac{1}{3}$ unit,

(b) by limiting the range of sizes considered at each stage of the procedure. This can be used to accentuate the effects produced by the ordering methods of (a) above. If, for example, in option (i) above we not only wanted to ensure the inclusion of large sizes but also the complete exclusion of small ones, then only sizes greater than a certain minimum would be input to the pattern enumeration routine, e.g. if the only sizes included are those greater than 2000 the pattern selected is:

1 of 5900

1 of 4000

1 of 2100

compared to:

1 of 5900

1 of 5075

1 of 725

1 of 300 previously.

This restriction may also be used very effectively with the demand ordering to guarantee long production run patterns. A slight variation of this is, instead of fixing a cut-off point on the list, to start with a list containing only one order size and then increase the number in the list until an acceptable pattern is found. This would yield the pattern:-

3 of 4000

when ordered by descending size as above.

(c) by limiting the number of times a size may appear in a pattern. This restriction can be used when considering an integer problem to ensure that not more than the quantity of the sizes required appears in the pattern generated. This would change the pattern generated by the ascending demand ordering, i.e. (a)(iv) from:

12 of 725

1 of 2100

1 of 1200

to:

4 of 725

3 of 2425

1 of 1225

2 of 300

and thus the amount to be cut is increased from $\frac{1}{3}$ to 1 which for an integer problem makes it a feasible pattern to cut;

(d) by limiting the number of pieces in a pattern. This can easily be achieved as shown in Program 2, Appendix 1.2, p.112, and is useful when there are a limited number of cutting edges. When this extra modification was added and the maximum number of pieces set at 6, then the above pattern was modified to:

3 of 725

1 of 3300

1 of 5075

1 of 1450

(e) by assigning a value to each pattern. This can be useful when the definition of an acceptable pattern is in terms of other things than trim; for example in a problem with high set up costs, the criterion for acceptability may be the number of order sizes whose remaining demand can be

simultaneously satisfied by the pattern. The enumeration is allowed to proceed until a pattern with an acceptable value is generated. This method tends to be very slow as a computation is required for each pattern.

The list above is obviously not comprehensive, but does illustrate the wide variety of patterns which may be selected from even a small number of order sizes by varying the enumeration technique.

4.5 Computational Efficiency

Most of the solution time in a pattern enumeration method will be taken up by the actual enumeration phase, it is thus important that this should be as efficient as possible. Several methods to improve the efficiency of the enumeration routine are given below. Each one has been run on a set of test data (Data set 1, Appendix 1.1, p. 111) to select the first pattern for the solution. No ordering methods have been applied.

(a) At each stage of the step

$$\text{"set } a_j = \lfloor \text{trim}/w_j \rfloor \quad \text{for } j = (\text{JP} + 1) \text{ to } M\text{"}$$

in Fig. 5 § 4.2, p. 27 a check is made whether the trim remaining is less than the smallest order size. If it is, then clearly all further a_j are zero and the program can proceed to the next step. This modification has been included in Program 3, Appendix 1.2, p. 113. The time taken to find an acceptable pattern was reduced from 28 computer secs. to 23 secs.

(b) A further refinement, similar to that above, is, instead of merely checking whether trim is less than the smallest size, to check whether trim is less than any of the sizes lower in the order list (Program 4, Appendix 1.2, p. 113). Computation time was now further reduced to 10 secs.

(c) Not untypically in scheduling situations, the vast majority of orders may have come from one source and have their demand sizes rounded, say, to the nearest 5 mm, while a small number of orders are for sizes rounded to only the nearest millimetre. Order lists of this kind can often lead to a very large number of patterns being enumerated before an acceptable one is found. The reason for this will become clear from the following simple example:

<u>Size</u>	<u>Current Pattern</u>
401	4
1 325	2
2 430	3
455	<u>1</u>

Trim of 1 on a parent size
of 12 000.

No matter how one combines pieces of size 1325, 2430 and 455, the result will always be a multiple of 5. Thus the trim of 1 can not be reduced until the number of pieces of size 401 is changed. The enumeration can thus be speeded up considerably if all the patterns between (4, 2, 3, 1) and the first pattern containing 3 of 401 i.e. (3, 8, 0, 0) are jumped. The pattern (3, 8, 0, 0) has a trim of 197 (= 39 x 5 + 2). Since the sum of any multiples (whether positive or negative) of 1325, 2430 and 455 will always be a multiple of 5, all patterns containing 3 pieces of size 401 will have a trim of at least 2 and can be jumped. A similar argument can eliminate patterns containing 2 or 1 pieces of size 401. This is clearly a very great saving in the number of patterns which have to be enumerated.

The method outlined on the simple example above can be generalised as shown in the flow diagram, Figure 6, using a general test for common factors among the order lengths. Program 5, Appendix 1.2, p.114 is a complete program using this method. The method reduced the run time on the test data to less than 1 second. More generally, it has proved to be a most powerful method for reducing run times, e.g. the case study of Chapter 7;

(d) When using the modification of the enumeration approach so that a value is attached to each pattern (\S 4.4, p.39), then a "look ahead" procedure may be included to eliminate the enumeration of patterns which will not yield a higher value than that already obtained. A particular example of this is the procedure adopted by Gilmore and Gomory to solve the knapsack problem within their trim loss method (1C);

(e) When the option of increasing the size of the list (\S 4.4, p.39) is used there is obviously substantial repetition in the patterns generated. This may be overcome by introducing the extra demand size at the start of the list and

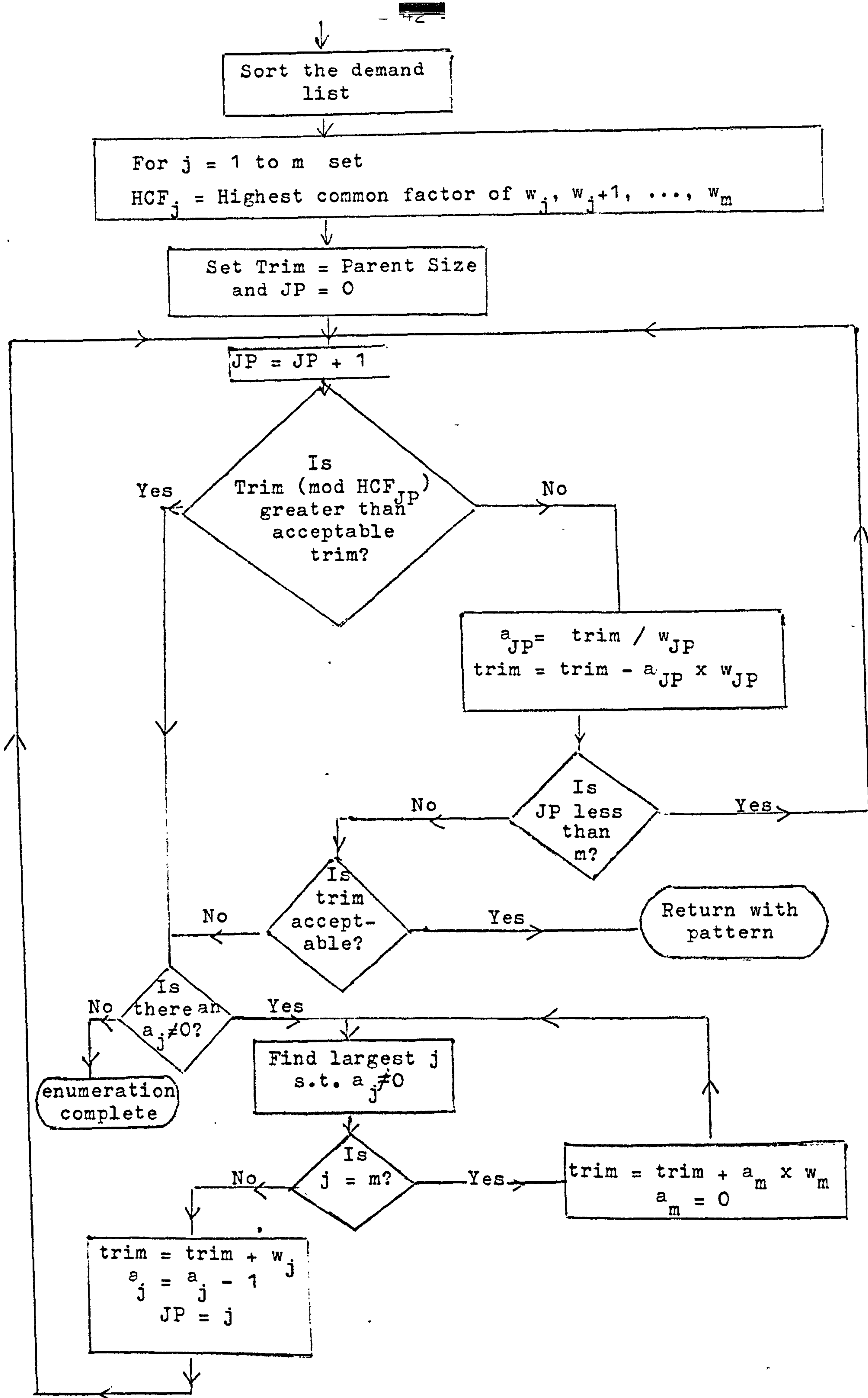


Figure 6. Flow diagram for Pattern Enumeration with a test to skip patterns which must be unacceptable as a result of common factors among sizes in the demand list.

terminating the search as soon as there is none of the newly introduced size in the pattern (Program 6, Appendix 1.2, p.115);

(f) Typically the pattern enumeration routine will be entered many times within a solution procedure with only minor changes to the list of sizes to be considered. This can be seen clearly in the worked example of §4.3. A saving of overall solution time can thus be made if, instead of entering the pattern enumeration routine at the beginning of the enumeration on each occasion, it is re-entered at the point where it was left. This method is used in the case study of Chapter 6 in Program 11, Appendix 3.2, p.148;

(g) Varying the acceptable parameter trim. This can be used to cut down the time spent searching for a solution by entering the enumeration routine with a very low acceptable trim parameter and then increasing it slowly as the enumeration proceeds without finding an acceptable pattern. It is, however, necessary to keep a record of the best pattern enumerated so that each time the acceptable trim is increased, a test can be easily made that a previously generated pattern would not have satisfied the relaxed trim constraint..

Generally, the above methods of reducing run times become more important as the complexity of the problem increases (normally with an increased number of sizes on the order list).

4.6 Introduction to the case studies

In order to test the ease with which pattern enumeration methods can be built and to assess their effectiveness, three case studies have been undertaken. These are described in chapters 5, 6 and 7. The case studies contain a wide range of the characteristics encountered in practical trim-loss problems. Each of the case studies is quite different from the others. They can therefore be used to give an indication of the range of problems which can be tackled by pattern enumeration methods.

In Chapter 5 a roll slitting problem in the paper industry is considered. In this problem there are heavy costs associated with patterns which can only be used for short production runs. These costs constrain the solution produced to contain far fewer patterns than there are ordered sizes. The solution required is very similar

to that had there been high set-up costs. The problems in Chapters 6 and 7 are both bar cutting problems in the steel industry with minimal set-up costs. That of Chapter 6 is a basically simple problem with no extra constraints or costs over the classical formulation, the only variation being that the scheduler is allowed to produce off-cuts. The problem of Chapter 7, on the other hand, is a highly specialized one containing important differences from the classical trim loss problem.

For each case study, a pattern enumeration method is developed step by step, showing exactly how the techniques of this chapter can be used to build a solution procedure for a practical problem. Although the final methods differ, the steps used to build them are the same i.e. it has been possible to adopt a common approach to three quite different problems.

The pattern enumeration programs developed for the case studies have been tested on the University of Hull's ICL 1905E computer. All reference to storage requirements and run times should be understood to refer to this machine.

5. A ROLL SLITTING PROBLEM

5.1 The Scheduling Problem

The selection of patterns for the reduction of parent widths to meet customer requirements is an important problem in the paper industry. Whilst circumstances differ from company to company, paper generally has a low value and thus scheduling methods which seek to minimize trim loss without reference to other production costs are unacceptable. The particular problem considered in this chapter is a good example. It has many features which render the application of classical optimization techniques difficult, viz. the limited capacity of the main slitting machine, the cost of reprocessing on a secondary machine, the limitation on the number of cuts per pattern, the requirement for integer solutions, and the existence of tolerances on over and under production. As such it provides a suitable test case for heuristic methods.

The problem has previously been considered by Haessler and the details of the problem are drawn exclusively from his papers [(11), (12)]. Paper is cut to meet customer orders and not for stock. Orders are for an integer number of rolls of specified widths. There are tolerances on the quantities ordered i.e. over and under production is permitted within certain limits. Typically the scheduler is required to satisfy the demand for twenty different widths, and the average ratio of parent width to finished width is 7:1. The actual production sequence is shown in figure 7 below.

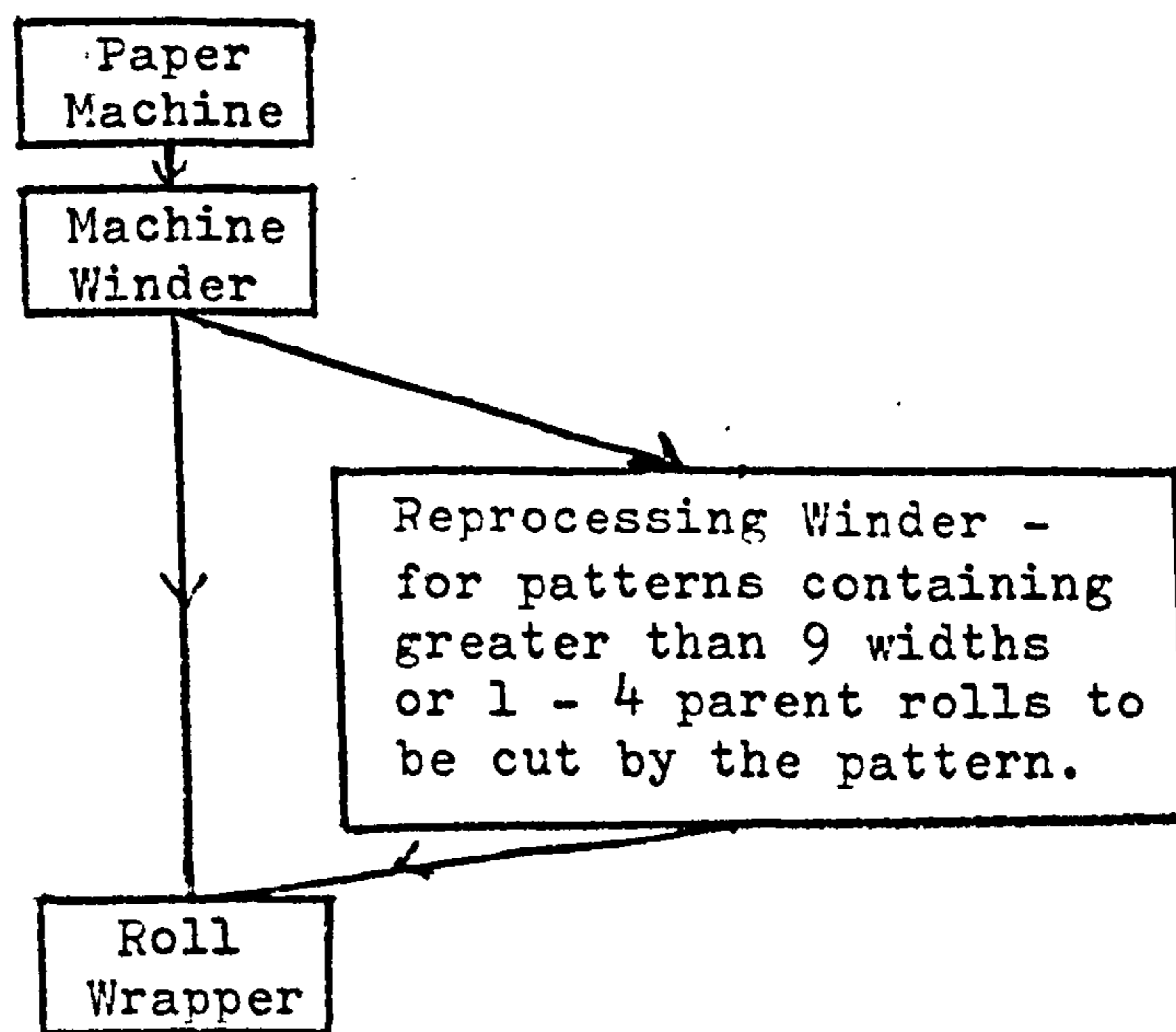


Figure 7. Production Sequence.

A continuous band of paper is produced by the paper machine. It takes approximately 16 mins. to produce sufficient for a 36" diameter roll. Before being rolled, however, the band of paper is slit along its length by the machine winder. The set-up time for the machine winder is such that a pattern can be set up and three rolls cut in about 45 mins., whilst the actual paper required would be produced by the paper machine in about 48 mins. In view of the inevitable contingencies which arise and the limited storage space between the paper machine and the winder, it is considered inadvisable to cut less than four rolls on the winder. When it is necessary to cut less than four rolls of a pattern then some of the work will be done on the reprocessing winder.

There are ten cutting edges on the machine winder, two of which are used to trim the edges. Thus if greater than nine widths are to be cut some of the slitting is done on the reprocessing winder.

5.2. Detailed Problem Characteristics

The high cost of stopping the paper making machine, the limited storage space before the winder, and the limited cutting capacities of both the machine winder and the reprocessing winder give rise to the following practical operating rules:

- (a) Patterns comprising no more than nine widths and to be used for at least five rolls are cut entirely on the machine winder;
- (b) Patterns comprising ten or eleven widths to be used for at least five rolls, and all patterns to be used for four rolls are only half set and cut on the machine winder. The remaining widths are cut on the reprocessing winder. This effectively reduces the set-up time on the machine winder and leaves a reduced parent roll of small enough width to be accommodated on the reprocessing winder;
- (c) All other patterns are slit down the middle on the machine winder and other slitting done on the reprocessing winder.

The parent band of paper produced by the paper machine is 200" wide; however $1\frac{1}{2}$ " must be left at each side so that the

rough edges may be trimmed, reducing the maximum usable width to 197". For each roll which is to be reprocessed, an extra trim of $\frac{3}{4}$ " on each side must be allowed. The reprocessing winder will slit rolls of up to 110" (i.e. $108\frac{1}{2}$ " maximum usable width). As with the machine winder, the reprocessing winder has a limited number of cutting edges - eight, so only seven widths may be cut at once. Thus if a roll is to be slit into greater than seven widths on the reprocessing winder it must first be pre-slit on the machine winder so that each roll going to the reprocessing winder is cut into at most seven widths. This procedure inevitably produces extra trim as each roll reprocessed incurs a minimum additional trim of $1\frac{1}{2}$ ". Patterns of this type are therefore avoided whenever possible.

Haessler assesses the quality of a cutting schedule in terms of the trim incurred (including the side trim on the reprocessing winder, but not the unavoidable side trim on the machine winder) and the number of rolls to be reprocessed. The cost of reprocessing a roll is calculated to be \$15. The value of the trim loss is \$135/American ton and the rolls considered weigh 28 pounds/inch of width; thus the cost of trim is \$1.89/inch/roll.

It is technically possible to implement a schedule requiring a non-integer number of rolls to be cut by a pattern, as rolls can be spliced. However the high cost of splicing compared to the paper cost for the grades under consideration rules out splicing (and hence non-integer solutions) for practical purposes.

Haessler requires the quantities of each size scheduled to lie within the tolerances shown below in Table 2. These tolerances are somewhat more restrictive than those used in the industry in practice, but will be used within this case study so that direct comparisons can be made with Haessler's results. No penalty is attached by Haessler to either over or under production.

Table 2. Tolerances allowed on order quantities.

Quantity ordered (rolls)	Tolerance (rolls)
1 - 19	0
20 - 39	± 1
40 - 99	± 2
100 - 199	± 3
200 +	± 4

5.3 Mathematical Formulation

The complete problem may be expressed in mathematical form as:-
 Given a demand for d_j rolls of width w_j with a tolerance of $\pm t_j$ rolls
 ($j = 1, 2, \dots, m$),
 define a_i ($i = 1, \dots, n_1$) as the i^{th} pattern which can be cut
 entirely on the machine winder and x_i as the number of rolls to be
 cut to the i^{th} pattern.

$$\begin{aligned} \text{Then } \forall i \quad \sum_j a_{ij} &\leq 9 \\ \sum_j a_{ij} w_j &\leq 197 \\ x_i &\geq 5 \end{aligned}$$

Similarly define b_i ($i = 1, 2, \dots, n_2$) as the i^{th} pattern which can
 be half-cut on the machine winder and y_i as the number of rolls to
 be cut to the i^{th} pattern.

$$\begin{aligned} \text{Then } \forall i \quad \sum_j b_{ij} &\leq 11 \\ \sum_j b_{ij} w_j &\leq 195.5 \\ y_i &\geq 4 \end{aligned}$$

Also define c_i ($i = 1, 2, \dots, n_3$) as the i^{th} pattern which must be
 slit completely on the rewinder and z_i as the number of rolls to
 be cut to the i^{th} pattern.

$$\begin{aligned} \text{Then } \forall i \quad \sum_j c_{ij} w_j &\leq 194 \\ z_i &\leq 3 \end{aligned}$$

$$\begin{aligned} \text{Overall we require } \sum_i a_{ij} x_i + \sum_i b_{ij} y_i + \sum_i c_{ij} z_i &= P_j \\ d_j - t_j &\leq P_j \leq d_j + t_j \end{aligned}$$

where all variables take integer values

and we wish to minimize

$$Z = 1.89 \left(197 \left(\sum_i x_i + \sum_i y_i + \sum_i z_i \right) - \sum_j P_j w_j \right) + 15 \left(\sum_i y_i + 2 \sum_i z_i \right)$$

i.e. minimize total costs.

In the formulation above, patterns requiring three or more rolls
 to be cut on the reprocessing winder have been excluded so that the
 structure of the problem is not clouded by excessive detail. These
 patterns can easily be added to the formulation by defining additional
 sets of pattern vectors.

5.4 Current Scheduling Method

Different scheduling methods are considered by Haessler (11).
 Mathematical programming is considered unsuitable as the solutions

obtained contain many patterns which are only used for relatively short run lengths and the consequence of short run lengths is an excessive amount of reprocessing.

Given that typically for this problem there will be about twenty different widths to be scheduled, and that the average ratio of parent width to ordered width is 7:1, there are a very large number of possible patterns (see Table 1, p.16). Branch and bound methods are rejected by Haessler because of the prohibitively long search times which arise whenever a large number of patterns must be examined.

Having found the heuristic methods of Pierce (16) and Johns (13) to be not quite satisfactory, Haessler developed his own heuristic procedure. Haessler's method (see Fig. 8) recognises three types of pattern:

- (a) Those which can be cut entirely on the machine winder;
- (b) Those which can be half cut on the machine winder;
- (c) Those which must be cut entirely on the reprocessing winder.

At each stage of the iteration procedure, the type of pattern sought, the number of rolls to be cut, the minimum and maximum number of cuts in the pattern, and the permissible trim loss are determined by the current values of two descriptors of the residual problem, i.e.

(a) The estimated number of rolls required to satisfy residual demand ($\sum_j d_j w_j / 197$).

(b) The average number of widths in each pattern ($197 \sum_j d_j / \sum_j d_j w_j$).

The residual problem descriptors are used to determine the type of pattern to be sought and 'aspiration levels' i.e. constraints on pattern usage, number of cuts, and trim loss. The required usage is reduced until a pattern satisfying the other aspiration levels is found or until it becomes necessary to change the pattern type sought.

The pattern enumeration technique used by Haessler differs very little from the basic procedure described in Chapter 4. Demand widths are input into the generating procedure in descending order of residual demand. The only modifications to the basic enumeration procedure are those necessary to ensure that the size with maximum remaining demand appears at least once, and that no width appears

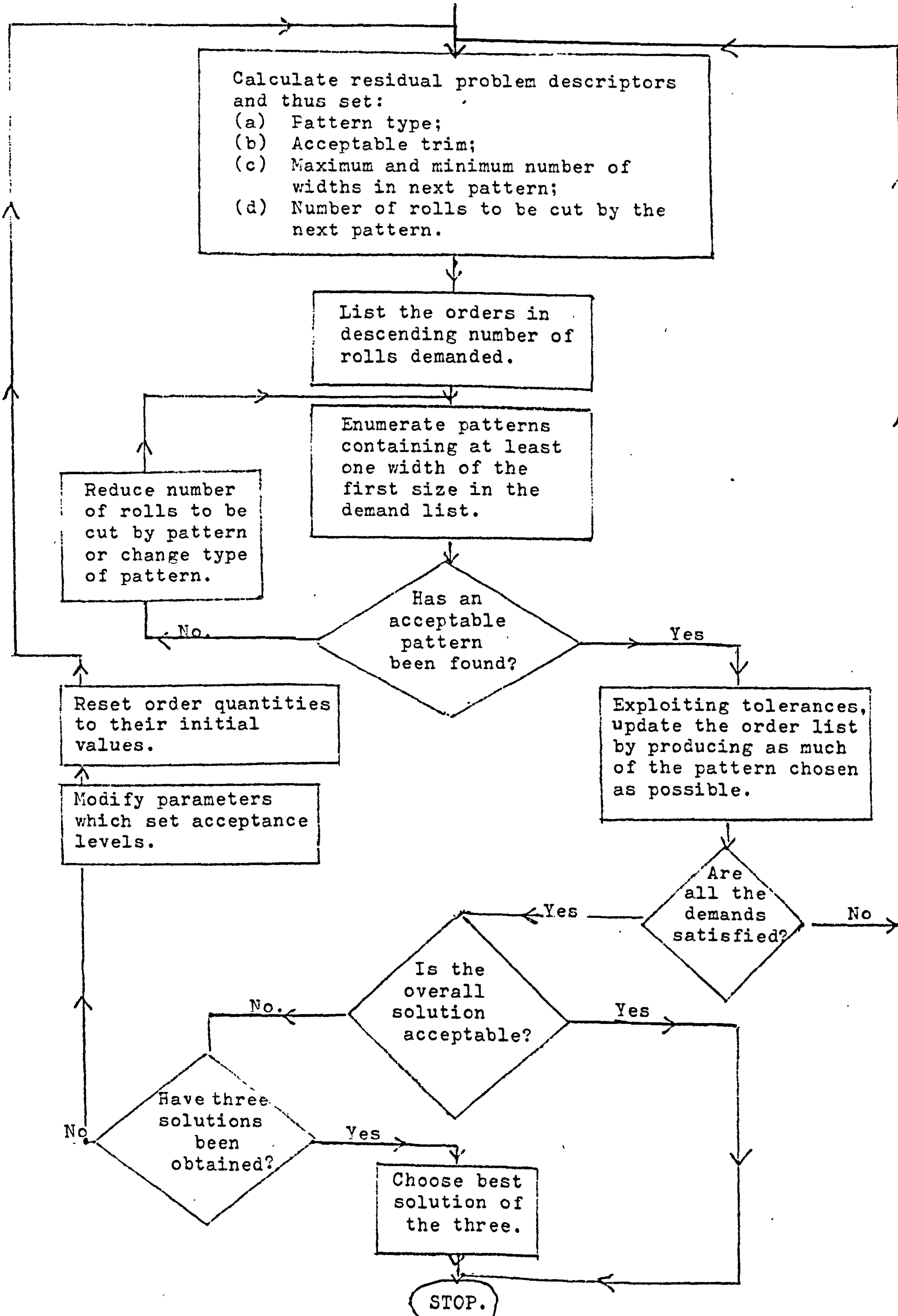


Figure 8 Flow chart of Haessler's method of solution to the paper slitting problem.

more frequently than residual demand allows, given the predetermined run length for the next pattern. Patterns which fail to meet the current aspiration levels for number of cuts or trim loss are simply rejected.

When an acceptable pattern has been found, the tolerance limits on demand for widths in that pattern are exploited in an attempt to satisfy simultaneously total residual demand for more than one width.

The above procedure is described as a single pass heuristic. Having found that the schedules obtained were not always satisfactory, Haessler devised a multiple pass heuristic in which over-all aspiration levels for trim loss and number of rolls for reprocessing were set. If these levels were not met in a single pass then the control parameters for trim loss, minimum and maximum number of cuts, and pattern usage were modified and the process repeated up to three times. If no satisfactory solution was obtained after three passes then the best solution would be used.

Haessler develops his method on a set of twenty problems (shown in Appendix 2.1, p. 116 Data sets 2 - 21) for which manually generated schedules were available for comparison. He then validates the method on a further fifteen problems for which he unfortunately does not give data. A restriction on the data which can be used for comparison is that for nine data sets Haessler gives summary solutions which seem inconsistent with the data for one of two reasons:

- (a) The total amount produced (i.e. demand + net over-production) is not equal to the amount scheduled (i.e. Nos. rolls x 197 - Trim);
- (b) The net over-production could not be made up from the widths on which tolerances were allowed.

These nine sets of data have been rejected and the reason shown in detail with the data in Appendix 2.1, p. 116. When the sets of suspect data have been eliminated, eleven data sets are left (i.e. 2, 3, 5, 6, 7, 13, 14, 15, 17, 19, and 20) which can be used for comparison to evaluate any new method. Haessler's results on these problems is shown in Tables 20-23, Appendix 2.3, pp. 129 & 130.

5.5 Proposed Method

Essentially Haessler's method is to define three types of pattern and to modify aspiration levels for each according to the current

values of the residual problem descriptors. In the multiple pass procedure, parameters are changed if the over-all quality of the solution is not satisfactory. The actual pattern enumeration phase differs very little from the basic method outlined in Chapter 4.

It was felt that there was scope for improving the quality of the schedules obtained by using a pattern enumeration method which would generate patterns more suited to the particular requirements of this problem. If more suitable patterns could be generated, then the need for a multiple pass procedure would be avoided. This, coupled with faster enumeration using the methods of Chapter 4, could substantially reduce the amount of computer time required to reach a solution.

It was observed, that Haessler was obliged to sacrifice trim loss to a greater extent than was necessary because no attempt, other than by exploiting demand tolerances, was made to avoid creating intractable residual problems. In the method described below the pattern enumeration technique has been modified so as to avoid leaving residual demands that must necessarily be satisfied at extra cost by reprocessing. Tolerances are not exploited until the total residual demand requires less than one roll for completion. This ensures that customers' requirements are more exactly met.

The number of different types of pattern is expanded from three to seven in order to avoid intractable residual problems:

(i) Those patterns which can be used for six or more rolls requiring no reprocessing and which do not leave quantities of less than five rolls;

(ii) The remainder of those patterns which require no reprocessing;

(iii) Those patterns which can be used for five or more rolls requiring only one roll/parent roll to be reprocessed and which do not leave quantities of less than four rolls;

(iv) The remainder of those patterns which require only one roll/parent roll to be reprocessed;

(v) Those patterns which can be used for two or more rolls requiring two rolls/parent roll to be reprocessed;

(vi) Those patterns which can be used for two or more rolls requiring three rolls/parent roll to be reprocessed. (In practice no more than three rolls/parent roll ever need to be reprocessed as the demand sizes are sufficiently large);

(vii) Those patterns which will only be used to cut one parent roll.

At each stage only patterns of the type required are generated by the pattern enumeration technique, i.e. the enumeration technique is automatically constrained to produce only patterns which can be used to cut the set number of rolls and which require a number of cuts within the limits imposed by the pattern sought. It is further constrained in stages (i) and (iii) not to produce patterns which will necessarily lead to reprocessing. Other modifications of the type described in chapter 4 are also included to improve computational efficiency.

The pattern enumeration method for (i) - (vi) outlined above is shown in the flow diagram (Fig. 9) below. The method for (vii) is slightly different as the tolerances can be exploited at this stage to reduce the trim loss on these patterns which would otherwise be large as a result of the small choice of feasible patterns. The method for (vii) is shown in the flow diagram, Fig. 10. The relaxation of the trim loss constraint at the end of each phase is a further attempt to avoid the more costly reprocessing incurred when a "lower order" type of pattern is used.

When programmed (program 7, appendix 2.2, p.125), the method described above was used to schedule the test problems (Data sets 2, 3, 5, 6, 7, 13, 14, 15, 17, 19 and 20), the overall cost of the schedules produced was \$4433 compared to \$4834 for the manual method, and \$3855 for Haessler's multiple pass heuristic method (more detailed results are given in Table 24 Appendix 2.3, p. 131). The poor performance of the pattern enumeration method in relation to Haessler's method was a result of the very high cost of the solution obtained from Data set 17 which was caused by the large number of rolls of a particular width still to be scheduled late in the solution procedure. This deficiency could be corrected by relaxing the acceptable trim parameters for stages (i) - (vi) which were initially set arbitrarily at 0.5", 1.0", 0.5", 1.0", 2.0" and 2.0".

Another deficiency of the schedules could be seen from an analysis of the patterns produced. Some of the reprocessed rolls had arisen as a result of only narrow widths remaining late in the solution routine. These narrow widths could not be combined to provide a low trim pattern without using greater than 10 cuts and hence incurring reprocessing.

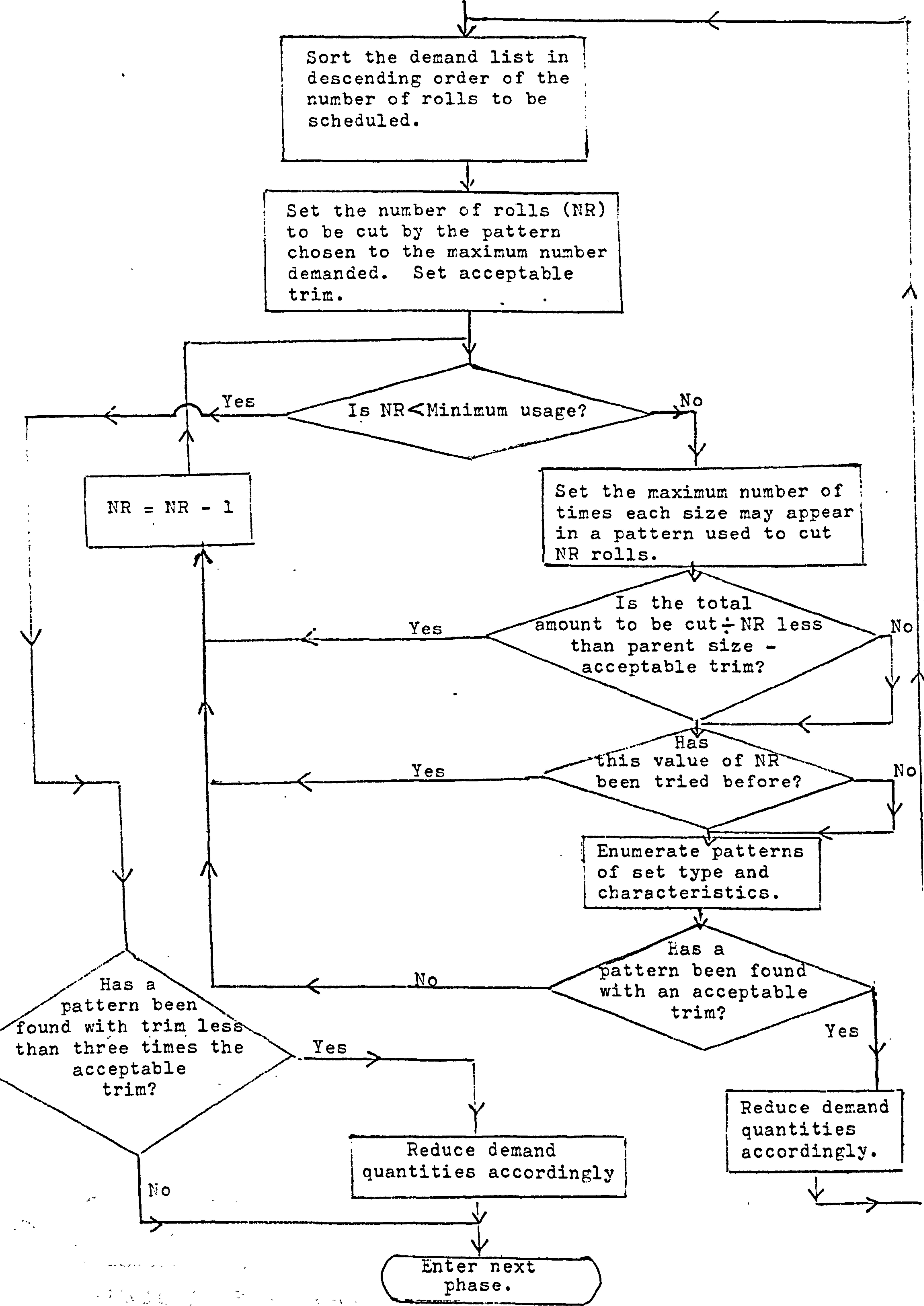


Figure 9 Flow diagram for the basic enumeration routine for the paper slitting problem (phases (i) - (vi)).

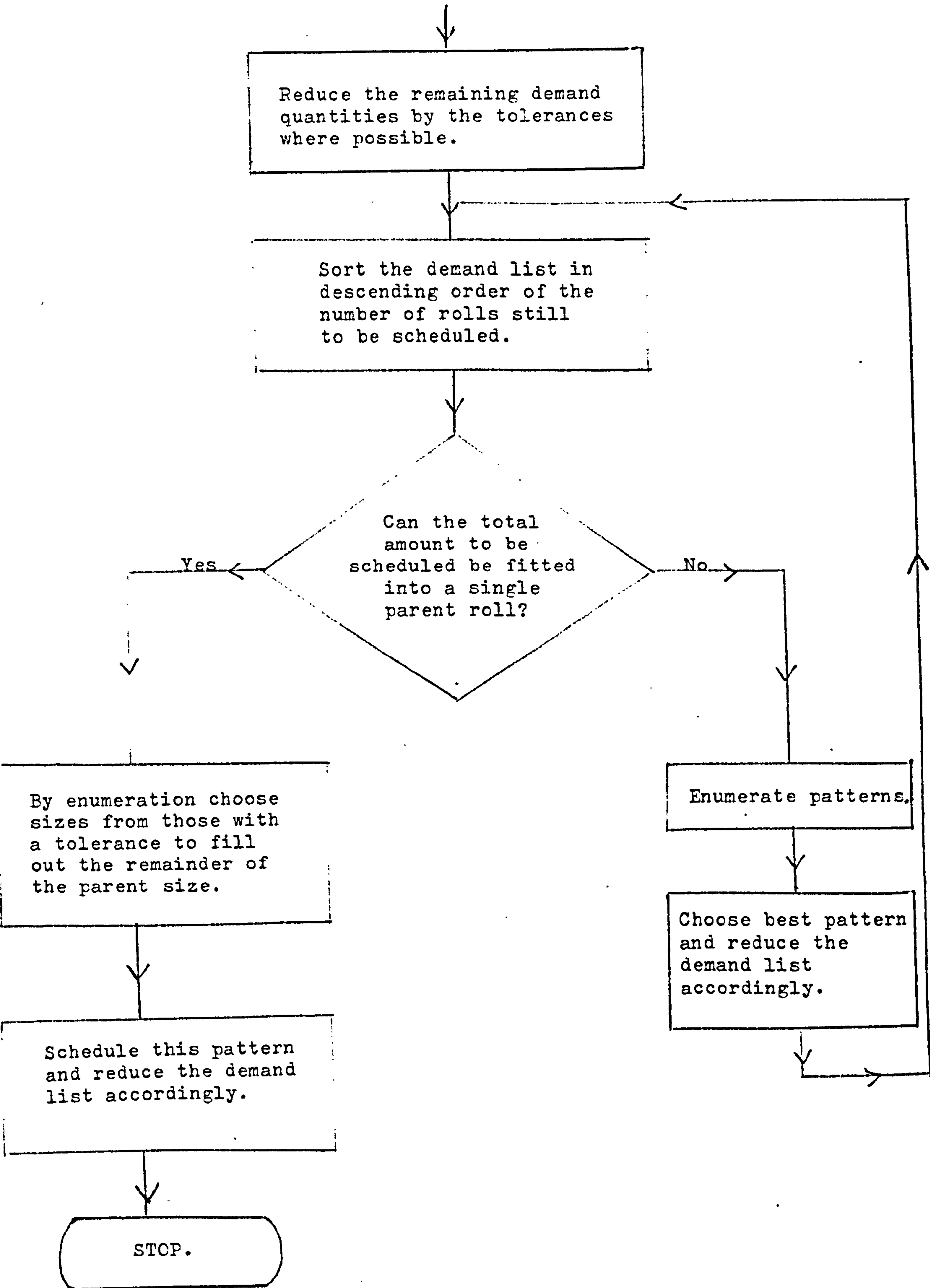


Figure 10 Flow diagram for the paper slitting problem to schedule remaining demands, exploiting the tolerances (phase vii).

This weakness of the method can be overcome by ensuring that the large widths are not used up too quickly. An extra test on the number of widths in the pattern to be scheduled at each stage was added to achieve this end, namely:-

Average number of pieces/pattern after the current pattern has been scheduled. \leq Maximum number of pieces/pattern in this phase.

Define d_j ($j = 1, 2, \dots, m$) to be the remaining demand for size w_j
 W the usable parent width
 L the maximum number of pieces/pattern in this phase.
 X the number of rolls to be cut by the current pattern
 T the trim on the current pattern
 N the number of pieces in the current pattern
 A the average trim in subsequent patterns

Then:

$$\text{Total amount left to be scheduled after the current pattern} = \sum_j (d_j w_j) - X (W - T)$$

$$\text{Total number of order rolls to be scheduled after the current pattern} = \sum_j d_j - X \times N$$

We require that

$$L \geq \frac{\text{Average number of pieces/pattern after the current pattern has been scheduled}}{\text{Total number of order rolls to be scheduled after the current pattern}} = \frac{(\sum_j d_j - X \times N)(W - A)}{\sum_j (d_j w_j) - X(W - T)}$$

$$\text{i.e. } N \geq \frac{\sum_j d_j - (\sum_j (d_j w_j) - X(W - T)) \left(\frac{L}{W - A}\right)}{X}$$

This will be approximated to by:-

$$N \geq \frac{\sum_j d_j - L \left(\frac{\sum_j (d_j w_j)}{W} - X\right)}{X}$$

Thus it is necessary in sections (i) and (iii) of the method to only accept patterns with a certain minimum number of pieces in a pattern. (It is not necessary to make this check in stages (ii) (iv) and (v) as the maximum number of pieces in a pattern will be increased in the stage following each of them, while stage (vi) is only followed by the routine to schedule all the small quantities left.) These modifications have been included in the program. The acceptable trim parameters were not modified at this stage as this was best done

once the details of the solution procedure itself had been finalised. The introduction of the minimum number of pieces in a pattern to the method produced different solutions for Data sets 3, 19, and 20, resulting in a reduction of £136 in the overall cost (see Table 25 Appendix 2.3, p.131).

Having thus finalised the method itself, the parameters had to be experimentally 'tuned' to obtain a good set of schedules. The acceptable trim parameters were modified to 1.0", 2.0", 1.0", 2.0", 2.0" and 2.0". This reduced the cost for data sets 3, 5, 7, 14, 15 and 17, but increased that of data sets 2 and 19, resulting in a reduction in overall costs of £889 (see Table 26 Appendix 2.3, p.132). However there was still a high cost from data set 17. The parameters were again modified, to 2.0", 2.0", 2.0", 2.0", 2.0" and 2.0". The scheduling cost of problem 17 was now reduced to an acceptable level but at the expense of an increase in costs for problems 2, 3, 5, 13, 15 and 19; the overall increase in cost being £176. (See Table 27 Appendix 2.3, p.132). The previous set of parameters were thus selected as most appropriate.

The enumeration method developed above can easily be changed from a single pass method to a multiple pass. The change would be made by running the program with several different values of the acceptable trim parameters. These different values need not be pre-set but could be dependent on the nature of the solution obtained in previous passes (as Haessler does in developing his multiple pass method from the single pass method). A reduction in the acceptable trim parameters of one phase will generally reduce the amount of trim produced by that phase, but at the expense of an increase in the amount of unsatisfied demand passed to the next phase of the method, and hence eventually to an increased number of rolls to be reprocessed. An increase in the acceptable trim parameter will have the opposite effect, of increasing trim but decreasing reprocessing. The magnitude of the reduction in cost if a multiple pass procedure were used can be judged from the fact that:

(a) Haessler increases the improvement over the manual solution from 13% to 20% when he changes from a single to a multiple pass procedure with the same basic method;

(b) A pseudo-multiple pass method for the enumeration method can be obtained by considering the best answer produced by any of the

three settings of the acceptable trim parameters (see Col. I, Table 28 Appendix 2.3, p.133) in which case the reduction in overall costs is increased from 30% to 42%.

5.6 Discussion

One would not normally make a comparison between the results obtained in the previous section and other methods applied to the same data since the results of the pattern enumeration approach are obtained using parameters "tuned" on the sets of data under consideration. Unfortunately, there is no other set of data available for comparison purposes. Haessler, however, uses the sets of data to develop and test his method, thus some comparison can be made between the pattern enumeration method developed above and Haessler's method as both have had the opportunity to adapt to these data sets. The two sets of results are summarized below in Table 3.

Table 3. Summary of solutions to the roll slitting problem

	Manual	Single pass		Multiple pass	
		Haessler's	Proposed	Haessler's	Proposed
Trim (inches)	1708	1243	657	1182	638
Number of reprocessed rolls	107	123	120	108	108
Cost (¢)	4834	4195	3408	3855	2826

Using the most effective set of parameters in a single pass procedure of the type described above, the overall cost of the schedules was 30% less than those obtained manually. The best set of schedules obtained by Haessler's single pass method was 13% better than the manual solutions; furthermore, the extent of overproduction using Haessler's method is significantly larger than that using the proposed method. Haessler increased the performance of his schedules when using a multiple pass procedure from 13% to 20%. The corresponding improvement for the proposed method is from 30% to 42%. It is interesting to note that although the proposed single pass method has substantially improved the quality of schedule, it is still possible to further improve the schedules by a multiple pass procedure.

Despite the fact that both Haessler's method and that developed above are based on pattern enumeration, there is a significant

difference in scheduling efficiency between them. The efficiency of the method considered here was improved at each step in its development by consideration of the nature of the residual problems arising. It is this avoidance of intractable residual problems which must account for its good performance relative to Haessler's method. The extent to which such problems can be avoided is likely to be one of the key determinants of the effectiveness of any pattern enumeration method.

The proposed pattern enumeration method took 83 seconds of computer time to generate schedules for the eleven test problems, i.e. an estimated computer cost of £8 for a saving of the order of £1400 over the manual schedules. At least in this case study, then, it is unlikely that the cost of generating a set of schedules by pattern enumeration will outweigh the savings in production costs.

6. A BAR CUTTING PROBLEM

6.1 The Scheduling Problem

It is one of the contentions of this thesis that a pattern enumeration approach is inherently flexible and can, therefore, be used for a wide range of trim loss problems. In the previous chapter a pattern enumeration method was developed for the particular characteristics of a roll slitting problem in the paper industry. In this chapter a quite different problem is tackled. The approach is applied to a cropping problem in a company producing steel reinforcement bars for use in reinforced concrete.

The scheduling problem faced by the firm in question (Jones Reinforcements Ltd., a member of the Cohen 600 group) is far simpler than that considered previously. Standard lengths of steel bar are purchased by the company, cut and bent to meet customer orders, then dispatched. As the company's production process consists of so few stages it is essential to its success that each stage is as efficient as possible. In particular the management is concerned to use "good" patterns (i.e. those which have low trim and give rise to small stocks of off-cuts).

Unlike paper slitting problems where the cost of the raw material is small, steel is relatively expensive. The costs of material wastage and stocks tend to overshadow other costs. Set-up costs are particularly low, as all that is required when a cutting pattern changes is that the bar to be cropped is lined up with a different mark on the cropping table. In the paper slitting problem, on the other hand, the time taken to set up new patterns imposed a major restriction on the type of schedules which could be used. The bar cropping problem is further simplified by there being no additional restrictions upon the type of pattern to be used in schedules, e.g. a maximum number of pieces/pattern. Additional differences between the two scheduling problems are that whereas there was a tolerance on the paper roll order quantities but no off-cuts permitted, here order quantities are to be satisfied exactly but off-cuts may be stocked.

Thus the bar cropping problem requires a quite different scheduling method to that developed in chapter 5. A suitable method is developed and solutions obtained for comparison with the company's existing computer-based heuristic procedures.

Were it not for the possibility of stocking off-cuts, the bar cropping problem could be considered as an integer version of the classical trim loss problem, in which the objective is simply to minimize wastage. Even with off-cuts the cropping problem is sufficiently close to the classical problem that the opportunity was taken to compare pattern enumeration solutions with those from a modified linear programming method.

6.2 Detailed Characteristics

Three types of steel bar are purchased by the company (mild, high tensile and square twisted). For each type nine diameters of bar are stocked (6mm, 8mm, 10mm, 12mm, 16mm, 20mm, 25mm, 32mm, and 40mm). Thus there are twenty-seven classes of stock items.

Customer orders are for various lengths to be cut from several bar types and diameters. In order that all the cut bars for a particular order may be kept together, the company schedules each customer order individually. The scheduling procedure currently used by the firm yields quite low trim loss and requires few off-cuts to be returned to stock.

Since it is the company's practice to schedule customer orders individually, the number of different demand lengths for a particular type of bar and diameter in a schedule is low, typically about 13. If orders were pooled by bar type and diameter then there would be a greater variety of order lengths within each scheduling problem, enabling better patterns and a more efficient schedule in terms of trim loss and off-cuts to be found. However the company would be posed additional problems in assembling complete customer orders. The effect of "pooling" is examined later.

The number of different lengths required of one bar type and diameter, i.e. the number of different lengths in a schedule, varies considerably between orders. In a sample of 402 orders, 58% required ten or less different lengths to be cut of any particular class of bar, while 3% required more than 50. The average was 13. The distribution is illustrated in Fig. 11.

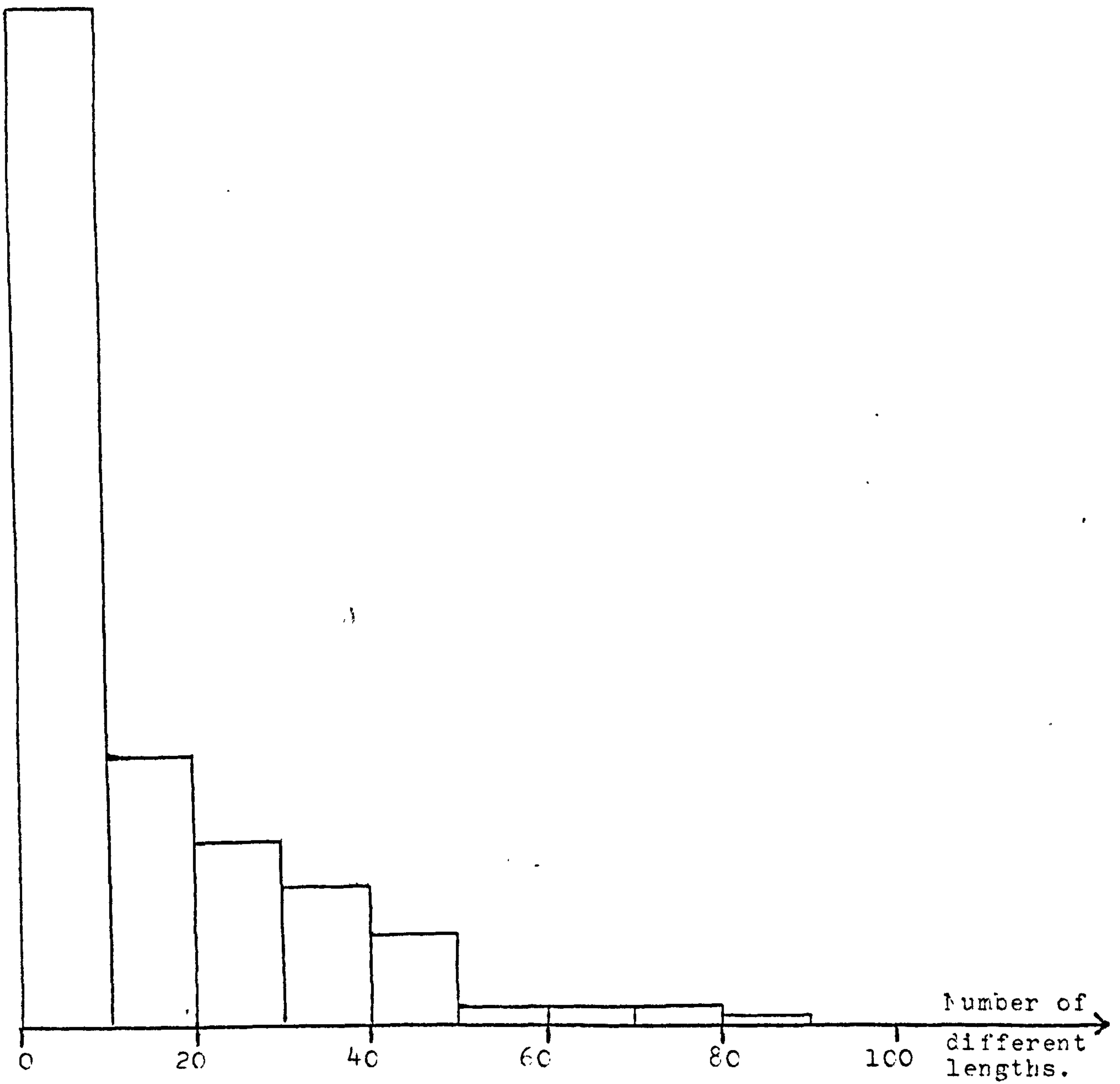


Figure 11. Histogram of the number of different lengths within a schedule.

53% of the demand is for sizes of less than 2 metres in length, and 0.6% exceed 10 metres. The mean length is 2.7m. The distribution is shown in Fig. 12.

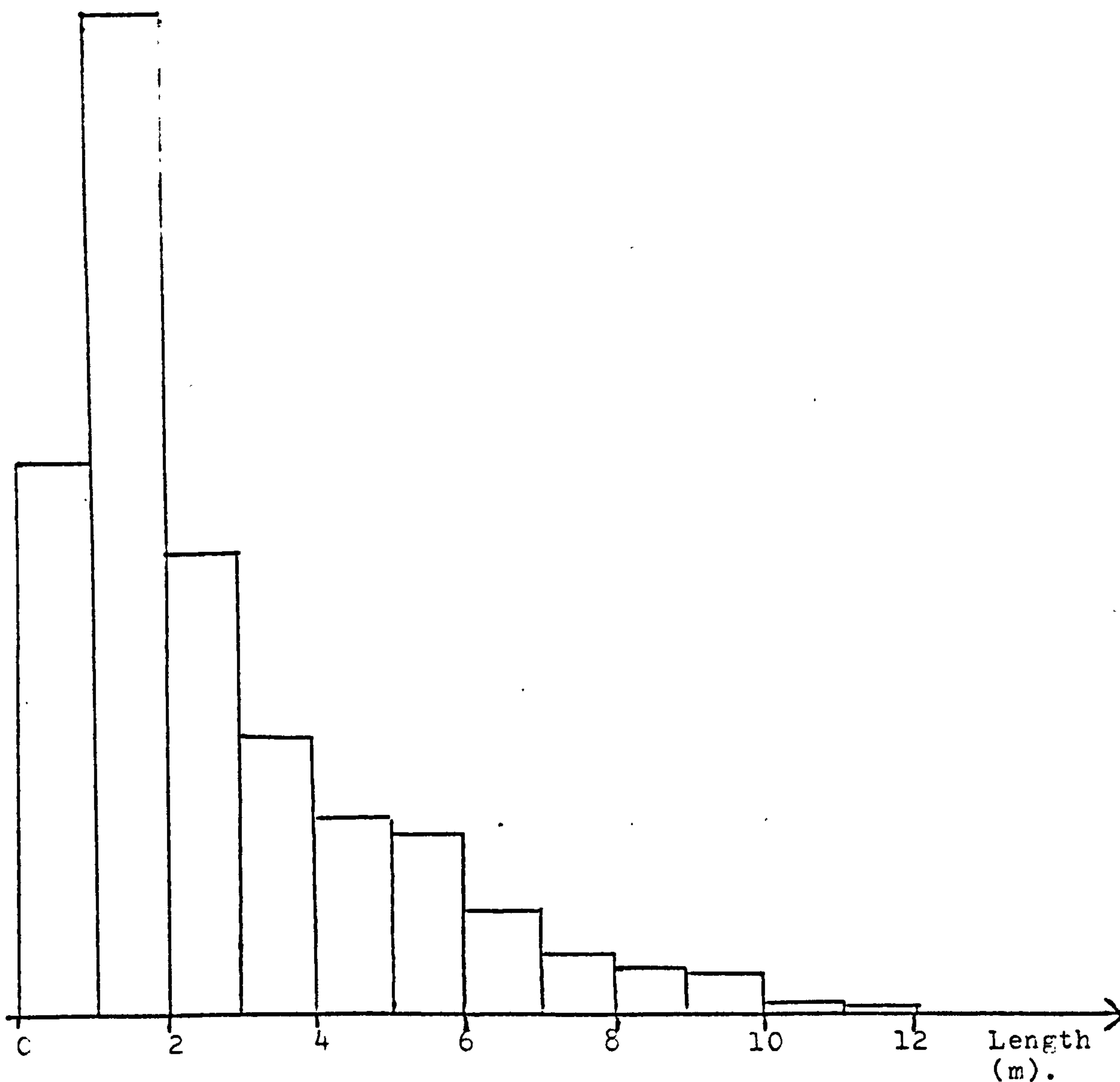


Figure 12. Histogram of the distribution of order lengths.

The stock lengths are 9m or 12m, and the distribution of orders gives rise to about four pieces being cut from each stock length. The patterns are, therefore, quite simple. The relatively small number of feasible patterns makes it difficult to avoid trim loss. Recognizing this fact, the company will stock off-cuts if this can reduce trim loss substantially. To ease stock-holding problems the company has decided that these off-cuts should be stocked as multiples of 20cm. and be longer than certain minimum lengths for each bar type and diameter. These minimum lengths have been set so that it will be relatively easy to use them in later schedules. (The shorter an off-cut, the more difficult it will be to find a pattern which uses it effectively).

6.3 Mathematical Formulation

Trim loss could be reduced to zero by returning all off-cuts to stock, but this would increase the quantity of stock held and, consequently, overall costs. In particular short off-cuts would tend to stay in stock for long periods of time before a cutting pattern which could use them was found. Some very short lengths may never get used. On the other hand, stock-holding costs could be reduced to a minimum by regarding all off-cuts as scrap, but such a policy would increase trim loss costs unacceptably. A compromise must be reached between these two extremes in such a way that the long term combined costs of trim loss and stock-holding are kept to a minimum.

Extending the notation of §1.6, the problem can be expressed mathematically as:-

$$\sum_{i=1}^n a_{ij}x_i = d_j \quad j = 1, 2, \dots, m$$

$$\sum_{j=1}^m a_{ij}w_j + t_i = L_k \quad \text{for some } k, \quad i = 1, 2, \dots, M$$

and we need to minimise

$$Z = \sum_{i=1}^n x_i f(t_i)$$

where: x_i are integer

L_k ($k = 1, 2, \dots, M$) are the lengths in stock

and $f(t) = \text{Min.} \left(\begin{array}{l} \text{cost of scrapping} \quad \text{estimated marginal stock} \\ \text{a piece of length } t, \text{ holding cost of a piece of} \\ \text{length } t \end{array} \right)$

In addition certain stock lengths may be in limited supply (possibly as a result of being off-cuts from an earlier schedule), in which case upper bounds will need to be placed on appropriate groups of x_i variables limiting the number of stock bars which can be cut using those patterns, i.e.

$$\sum_{i \in G_k} x_i \leq \text{Number of pieces of the } k^{\text{th}} \text{ stock length.}$$

where G_k is the group of patterns which can be cut from the k^{th} stock length.

Estimating the marginal cost of holding a piece of length t in stock would prove difficult as it will not only depend on t but also on the current stock position and the future order book. It was thus

impractical to calibrate the function $f(t)$ directly. Instead, as described below, a proxy has been used which gives acceptable results.

6.4 A Mathematical Programming Method

The formulation described above is close to that solved by Gilmore and Gomory [(9) & (10)] using linear programming. The two significant differences here being that integer solutions are required, and off-cuts may be sent to stock. The problem thus presents the opportunity to compare the performance of a linear programming-based method with other techniques, provided that the two differences between this and a standard formulation can be accommodated. This was achieved as follows.

If off-cuts of certain lengths may be held in stock, then patterns creating them can be brought into the Gilmore and Gomory method by defining a set of "dummy" parent lengths for each true parent length. Each dummy parent length is set to the parent length less one of the permissible off-cut lengths. The cost of using such a dummy length being the cost of material used plus the estimated cost of holding the resulting off-cut in stock.

In order to simplify computation a problem was chosen in which only one length of parent bar was available and no off-cuts were already in stock. (The existence of additional parent sizes or previous off-cuts which could be treated as parent sizes would not invalidate the method proposed but merely increase the number of feasible patterns to be considered). It was not possible to assess directly the cost of holding off-cuts in stock. Instead a cost function was assumed which could be tuned to change the relative cost of holding off-cuts to trim cost and also to change the relative cost of holding different lengths of off-cut in stock. Thus by tuning the cost function in different ways a range of solutions could be found and suitable ones compared with solutions produced by heuristic methods. The cost function adopted is :-

$$\begin{array}{l} \text{Cost of dummy parent size} \\ \text{of length } L \end{array} = L + \alpha L^\beta$$

where α and β are parameters to be tuned. The first term is a simple material cost, and the second the contribution of stock-holding. As α is increased, stock-holding costs will be increased relative to material costs, and thus patterns will be selected which are likely to create fewer off-cuts but more trim - the opposite effect being

created by reducing α . As β is increased, longer dummy lengths and their correspondingly shorter off-cuts will be more heavily penalized, and the average off-cut length will tend to rise. For all positive values of β , shorter off-cuts will be more heavily penalized than long ones as they will stay in stock longer.

The second difficulty to be overcome is the need for integer solutions. Gilmore and Gomory suggest that "given a non-integer solution one can round down to the nearest integer and treat the filling of the unfilled portion of the order as a separate problem to be solved by ad hoc methods". This approach has been followed, as shown in Figure 13.

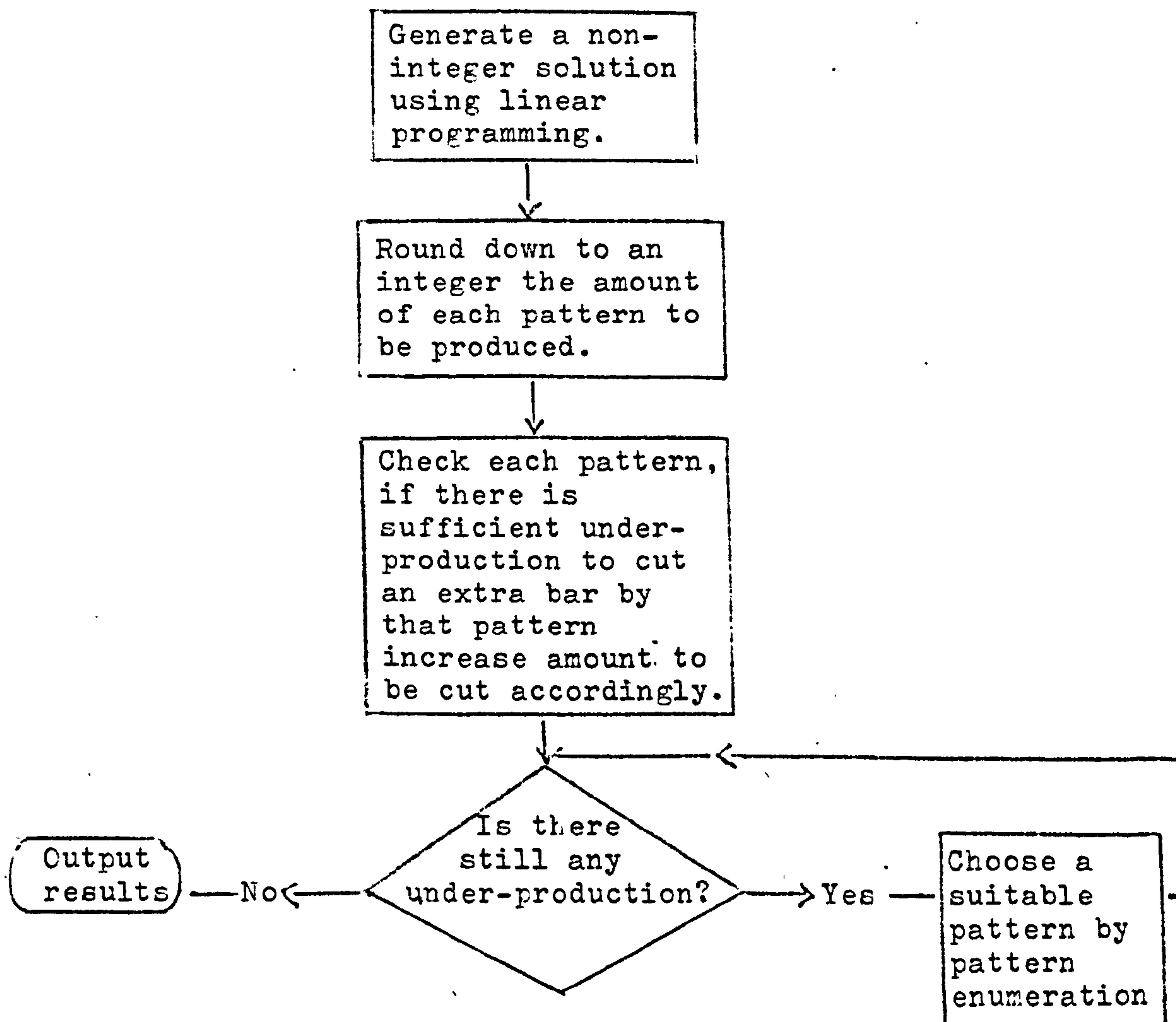


Figure 13. Flow diagram of the method used to move from a non-integer to an integer solution.

A program for the entire procedure including the linear programming phase was developed (program 8, appendix 3.2, p. 138). The program was used to obtain schedules for a particular problem* (data set 22, appendix 3.1, p. 134.) The program was run for several different values of α and β . The problem was also solved using the company's current heuristic (described in §6.5, p. 70) which was developed by Stainton, and a pattern enumeration method (developed in §6.6, p. 72).

Table 4. Comparison of schedules from different methods

Method	Total trim (in.)	Total length of off-cuts (in.)	Average off- cut length (in.)
Adapted Gilmore and Gomory	1 212	0	-
	934	3 960	220
	456	13 572	277
	410	14 967	217
	392	15 556	276
	344	18 484	279
	334	15 756	250
	223	2 205	78
	184	2 604	81
	145	2 763	83
	132	9 306	183
Stainton's method	480	12 108	224
Pattern enumeration	156	4 752	122

* The company had recently changed from imperial measurements to metric, so although the problem was described in metric measurements in §6.2 the test problems in appendix 3.1 are in inches.

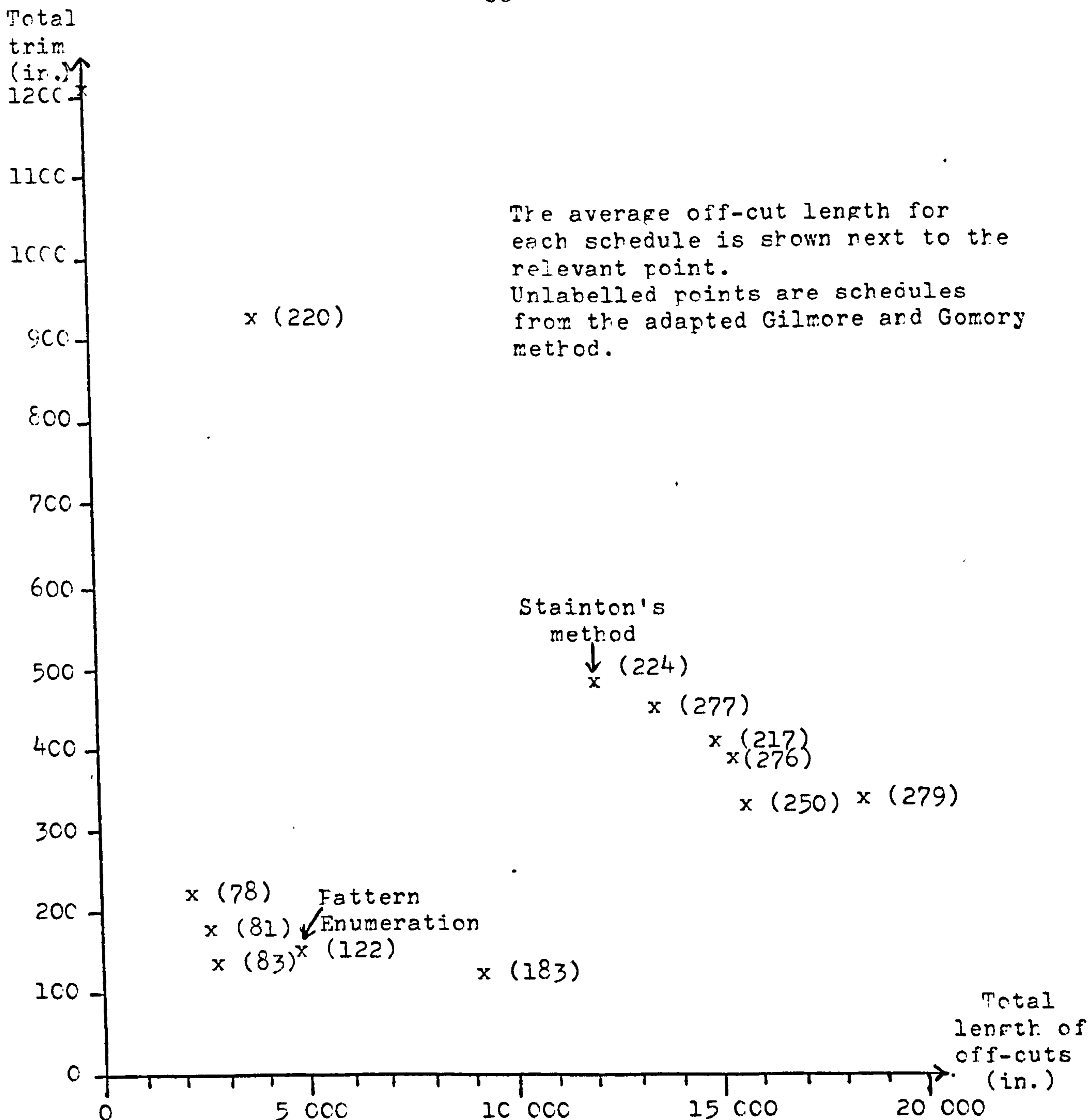


Figure 14. Trim loss and total length of off-cuts for different schedules.

Comparison of the different schedules is made difficult by the obvious inter-relationships between the three measures of solution quality. However the schedule from Stainton's method seems roughly comparable with the fourth IP solution, and the rattern enumeration schedule lies within the last group of IP solutions. It does not appear that the LF based method is performing significantly better than more simple techniques. What is more, the LF method takes considerably more computer time to produce a schedule. A similar result was observed when a similar exercise was carried out on data sets 23, 24 and 25.

Gilmore and Gomory observe, in relation to moving from non-integer to integer schedules by the type of procedure used here, that "since the cost first obtained is the smallest possible with or without the restriction to integers a small increase in it can often be tolerated even though the resulting cost may not be the best possible attainable with integers". If, as they imply, the non-integer solution is nearly optimal then the above exercise tends to suggest that the heuristic schedules must be equally close. If, on the other hand, they are mistaken and the rounding procedure moves the solution away from optimality, then one must question whether it was worth seeking an optimal solution in the first place. While it is impossible to draw hard conclusions on the basis of so limited a test, the effectiveness of pseudo-optimizing methods on this type of problem must be questioned.

The experience in designing heuristic methods, described in this thesis, shows that the nature of residual problems which are left after the bulk of the orders have been scheduled can have an important effect on solution quality. Great care must be taken when selecting earlier patterns that an intractable residual problem is not left. The case study of chapter 5 showed this particularly clearly. Heuristics are designed so that large awkward sizes will be scheduled early within any routine.

When heuristic methods are used to modify a linear programming solution to one satisfying all the problem requirements they are, in effect, solving a residual problem. In the method shown in Fig. 13, a residual problem is generated by rounding down the linear solution to integer values. Inevitably the residual problem will contain a mixture of the order lengths from the original problem, including awkward ones. Even an optimal solution to such a residual problem is likely to be costly. The only way to avoid such costs is to prevent such intractable residual problems arising.

Within a mathematical programming framework the only way to stop such residual problems arising would be to include all the relevant factors in the mathematical programme, i.e. to formulate the model as a full integer programme. However as Stainton (19) points out "the cost of the computing would be greater than any savings which might be achieved in steel utilization". To use integer programming methods is impractical on problems of this size and even to use linear

programming-based methods would be to incur significantly increased computing costs to obtain solutions which on the whole are likely to be little better than those obtained by more simple heuristic methods.

6.5 Current Scheduling Method

Until 1964 all scheduling was done manually. Since then the company has been using a suite of programs developed by Stainton and maintained by IBM(UK). Each day the details of orders to be cut the following day are transmitted to an IBM computing bureau. The relevant programs are run and cropping schedules to fulfil the orders are obtained. Not only are there programs in the suite to perform the scheduling but also ones to aid invoicing, produce acknowledgement notes, labelling, sales ledgers, market analyses, delivery routing, work scheduling and stock control. Thus the scheduling programs described here are but one part of the whole system.

A listing of the scheduling program could not be obtained, but the key steps of the procedure have been described by Stainton (19) and are outlined below. The procedure tackles each customer's order as a separate problem. This enables each order to be kept together in the cropping shed, thus reducing handling charges. (The cost penalty, in terms of extra trim loss and stock of off-cuts, incurred as a result of tackling orders singly is considered later in §6.6 p.72)

As each customer's order is read by the program, the required lengths are sorted by size within each material/diameter type. A two-stage scheduling procedure is then entered.

In the first stage, the demand for each size is met by patterns which are multiples of a single size. Only patterns with a trim loss below a preset maximum are accepted. The implicit preference order for selecting patterns for each order size is:-

- (a) patterns leaving no trim from an off-cut produced earlier in this customer order.
- (b) " " acceptable " " " " produced earlier in this customer order.
- (c) " " no " " " " from stock.
- (d) " " " " " a standard stock length.
- (e) " " acceptable " " an off-cut from stock.
- (f) " " " " " a standard stock length.
- (g) " " a useful off-cut from a standard stock length.

h. Other patterns from a standard stock length or off-cut having a trim larger than is normally acceptable.

This is a satisficing procedure which, except for those of type h, yields patterns with trim below a pre-set acceptable level. It is designed to keep down the number of new off-cuts created.

The second stage attempts to further reduce the number of off-cuts by matching each off-cut produced to a multiple of one of the order lengths. If a match is found, and if there are patterns made up of only the matching size, then the number of bars to be cropped by such a pattern is reduced to use up as many of the off-cuts as possible. This may leave a few of the matching size unscheduled. These are put into a residual pool. Once all possible matching has been performed stages one and two are repeated to schedule the sizes in the pool.

This method has been used satisfactorily by the company for many years. It currently gives rise to a trim loss of about 1% (approximately 250 tons/year) and an average stock of off-cuts of 400 tons. The company is able to adjust parameters within the method which control the acceptable trim loss and the sizes which may be considered as off-cuts. They can thus tune the method to yield schedules which observe a balance between reducing trim and reducing off-cut stocks that is appropriate to the market conditions they are facing. As the conditions change, so they can adjust the parameters and weight the two objectives differently.

For comparison with the pattern enumeration method developed in §6.6 it is worth noting the key elements of Stainton's heuristic method:

- (a) It is a satisficing procedure;
- (b) Patterns are accepted or rejected individually;
- (c) Its control parameters can be adjusted to suit changing conditions;
- (d) Sizes are listed by descending length;
- (e) The patterns used are made up of only one or two different sizes;
- (f) It is a multiple pass procedure.

Characteristics(a)-(c)are similar to those used in the pattern enumeration method developed below, while the methods differ in(d)-(f).

6.6 Proposed Method

The pattern enumeration procedure described in chapter 4 was used as the basis for a method to produce suitable cropping patterns. To develop an effective method for generating schedules it had to be built around the particular characteristics of Jones' scheduling problem. As with Haessler's problem it was a case of adapting the input listing, the types of pattern to be generated, and the pattern acceptance criteria, until no further improvement seemed possible. The way in which the procedure was developed and the quality of the schedules obtained at each stage is described below in order to illustrate not only how a pattern enumeration method may be built up, but also the effect of each adaptation.

An initial method was developed by building the simplest of structures around the basic procedure. Essentially the only enhancement required was to allow off-cuts to be returned to stock (see Fig. 15).

The schedules produced by the method will depend on the setting of the acceptable trim parameter. If it is set high then the trim on each pattern is likely to be high. On the other hand if it is set low then the trim on each pattern accepted into solution will be low but a difficult residual problem may be left, resulting in a large number of off-cuts. Also with a low acceptable trim parameter, computer time is likely to be long as many patterns will have to be generated before an acceptable one is found. It thus seemed reasonable to set the acceptable trim parameter low when there were many different order sizes still to be scheduled (i.e. when patterns satisfying a rigorous acceptance criteria were likely to be found) and to progressively relax it as the residual problem became smaller. Of the simple rules tested, the one below seemed to perform most satisfactorily.

$$\text{Acceptable trim} = \text{Max.} \left(0, 9 - \frac{\text{Number of different sizes with unsatisfied demand}}{2} \right)$$

A Fortran program was written for the procedure and schedules produced for 17 problems (Data sets 22-38, appendix 3.1, p.134). Each of these problems assumed a zero initial stock of off-cuts, simplifying the programming. The results of this test are summarized below in Table 5 and given in more detail in Appendix 3.3, p. 155.

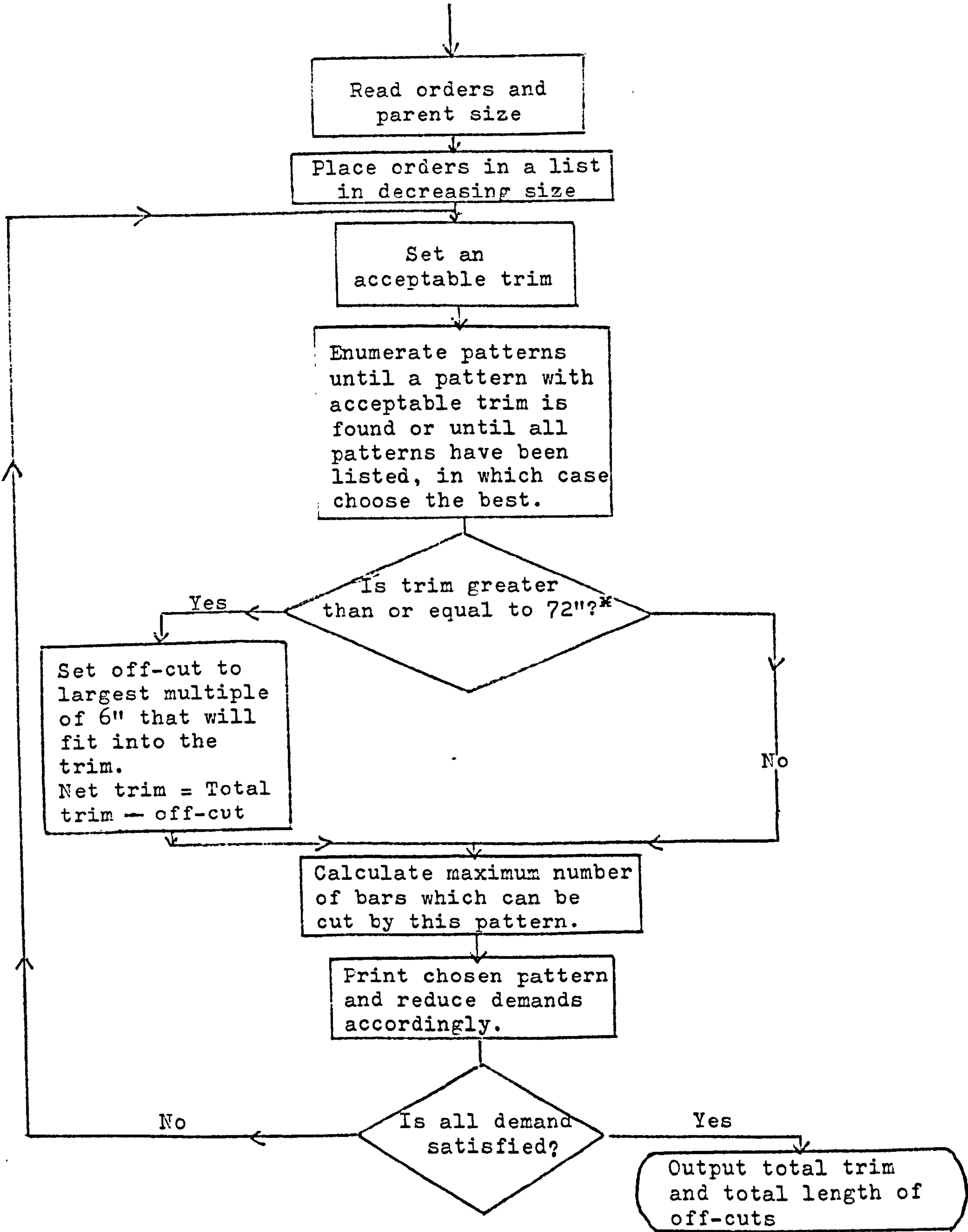


Figure 15. Basic Pattern Enumeration Method for the Bar Cutting Problem.

* For the diameters and types to be scheduled, 72" was the minimum size which the firm would consider sending to stock as an off-cut.

Table 5 Summary of the schedules from the first pattern enumeration method.

	Stainton's Method	Pattern Enumeration Method
Total Trim	11 545"	16 015"
Total length of off-cuts	192 030"	56 712"
Total number of off-cuts	1 057	378
Average length of off-cuts	182"	150"
Computer run time	Not known	16 secs.

These initial results, particularly the schedules themselves, and further consideration of the problem characteristics provided a basis for deciding how to adapt the method. It is clear, for example, that the pattern enumeration schedules give weight to reducing off-cuts (at the expense of trim loss) relative to the firm's schedules.

First it was noted that much of the pattern enumeration schedules trim loss was incurred by patterns having a trim in the range 10" to 71", these patterns being used to crop relatively few bars. The amount of trim could be considerably reduced, at the expense of increased off-cuts, if such patterns were replaced by ones having a trim large enough to be stocked as an off-cut rather than being scrapped.

Secondly, Jones' stock system is such that off-cut lengths are recorded in multiples of 6". An off-cut of length 125" would be recorded as 120", the remaining 5" being regarded as scrap. The initial pattern enumeration method does not take account of this characteristic. Its solutions can therefore be improved if those patterns creating off-cuts whose lengths are multiples of 6", are accepted in preference to other off-cut creating patterns.

The method was modified, as shown in Fig. 16, to cover these two points and a revised set of schedules obtained. The choice of a 10" limit on trim was arbitrary; had it been set higher then the amount of trim would have increased but at the expense of the number of off-cuts and vice versa had it been lower. The limit can obviously be tuned later once the solution method has been fixed.

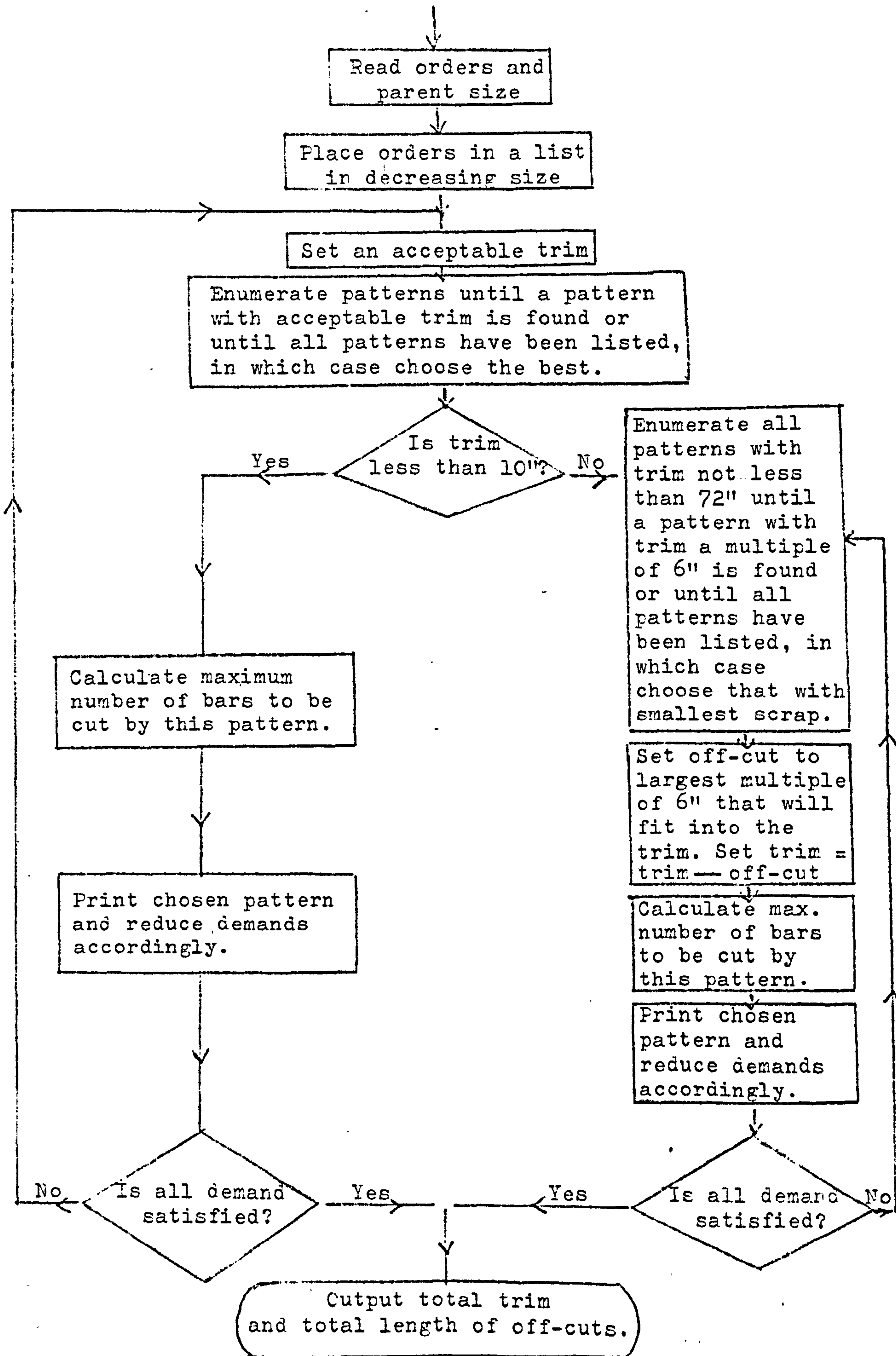


Figure 16. Modified Pattern Enumeration Approach for the Bar Cutting Problem.

Table 6. Summary of the schedules from the modified pattern enumeration method.

	Stainton's Method	Pattern Enumeration	
		Basic Method	Method modified to take advantage of the company's off-cut policy
Total Trim	11 545"	16 015"	4 507"
Total length of off-cuts	192 030"	56 712"	138 804"
Total number of off-cuts	1 057	378	844
Average length of off-cuts	182"	150"	164"
Computer time	Not known	16 secs.	16 secs.

In this form the procedure is a straightforward application of the basic pattern enumeration technique with no attempt to 'tune' either selection criteria or the enumeration order. Despite this, the results show a significant improvement over Stainton's method which is currently used by the firm. Both trim loss and off-cuts were reduced.

Experiments were then performed on the enumeration order by changing the input listings. In the basic procedure described above, order lengths were listed by size, and, as a result, longer lengths tended to appear in earlier patterns leaving shorter ones to later patterns. This yields efficient schedules provided that there is not a large number of one of the shorter lengths to be cut. If there are, then a residual problem requiring many pieces of one short length can prove impossible to schedule efficiently. On the other hand, if order lengths are sorted by the number of pieces still required of each length, then an equally intractable residual problem can result from only long pieces remaining to be scheduled. In order to obtain efficient schedules one must reduce the risk of generating a residual problem of either of these types. This will necessitate including both size and demand in the criteria used to sort the order list. Table 7 below shows that when the order list is sorted by the product of size and residual demand, then the schedules produced are better than those using demand alone but not as good as those using size alone. Detailed examination of the schedules showed that awkward residual problems of the second type discussed above were still

occurring. The ordering criterion was therefore modified to put greater weight on the avoidance of these residual problems by sorting the order list by the product of size and $\sqrt{\text{residual demand}}$. Program 9 in appendix 3.2, p.143 shows how the method was programmed.

Table 7. Summary of experiments to demonstrate the effect of different input listings.

	Stainton's Method	Pattern Enumeration Method			
		Sort by size	Sort by demand	Sort by size x demand	Sort by size x $\sqrt{\text{demand}}$
Total trim	11 545"	4 507"	4 687"	4 627"	3 883"
Total length of off-cuts	192 030"	138 804"	194 784"	155 628"	141 780"
Total number of off-cuts	1 057	844	1 245	941	822
Average length of off-cuts	182"	164"	156"	165"	172"
Computer time	Not known	16 secs.	28 secs.	24 secs.	28 secs.

The effect of different listings is significant but not dramatic. Whilst listing the residual sizes in a particular order increases the likelihood of a size high in the list being included in the next pattern, an awkward size could still be omitted, ultimately leading to highly inefficient patterns at the end of the scheduling procedure. The next modification, therefore, was to limit the search, at each stage, to only those patterns containing at least one piece of the first size on the list. With this limitation it is no longer necessary to set an acceptable trim higher than zero as the modification itself will ensure that awkward sizes do not get left.

The limitation on the set of patterns being searched should reduce computer run time. This can be reduced further by only sorting the order list once, rather than before each residual problem. It can be seen from Table 8 that run-times have been reduced by 25%. Trim has also been reduced, but at the expense of shorter off-cuts.

Table 8. Summary of schedules obtained when a limited pattern set is considered and the orders are only sorted once.

	Stainton's Method	Pattern Enumeration Method	
		Sort orders many times	Single sort and limited pattern set.
Total Trim.	11 545"	3 883"	2 143"
Total length of off-cuts.	192 030"	141 780"	104 520"
Total number of off-cuts.	1 057	822	876
Average length of off-cuts	182"	172"	119"
Computer time.	Not known	28 secs.	21 secs.

By now, as shown in Table 8, the improvements achieved over the company's method are considerable; an 80% reduction in trim and a 20% reduction in off-cuts. A further marginal improvement was obtained by taking advantage of a particular characteristic of Jones' order list. Frequently a size is ordered which can be cut exactly from the parent size (e.g. 36", 45", 72", 90" and 120" from a parent size of 360" in data set 29). As these can be easily scheduled the sorting procedure is adjusted to place these at the end of the order list. The flow diagram for the complete program (Program 10, appendix 3.2, page 146) with all the modifications included is shown in Figure 17. This final modification yielded schedules with a lower number of off-cuts at the expense of a slight increase in trim, as can be seen in Table 9.

Table 9. Summary of the schedules obtained by the final pattern enumeration method.

	Stainton's Method	Pattern Enumeration Method	
		Normal sort	Hold divisors of parent size at foot of order list.
Total trim	11 545"	2 143"	2 179"
Total length of off-cuts	192 030"	104 520"	101 604"
Total number of off-cuts	1 057	876	852
Average length of off-cuts	182"	119"	119"
Computer time	Not known	21 secs.	20 secs.

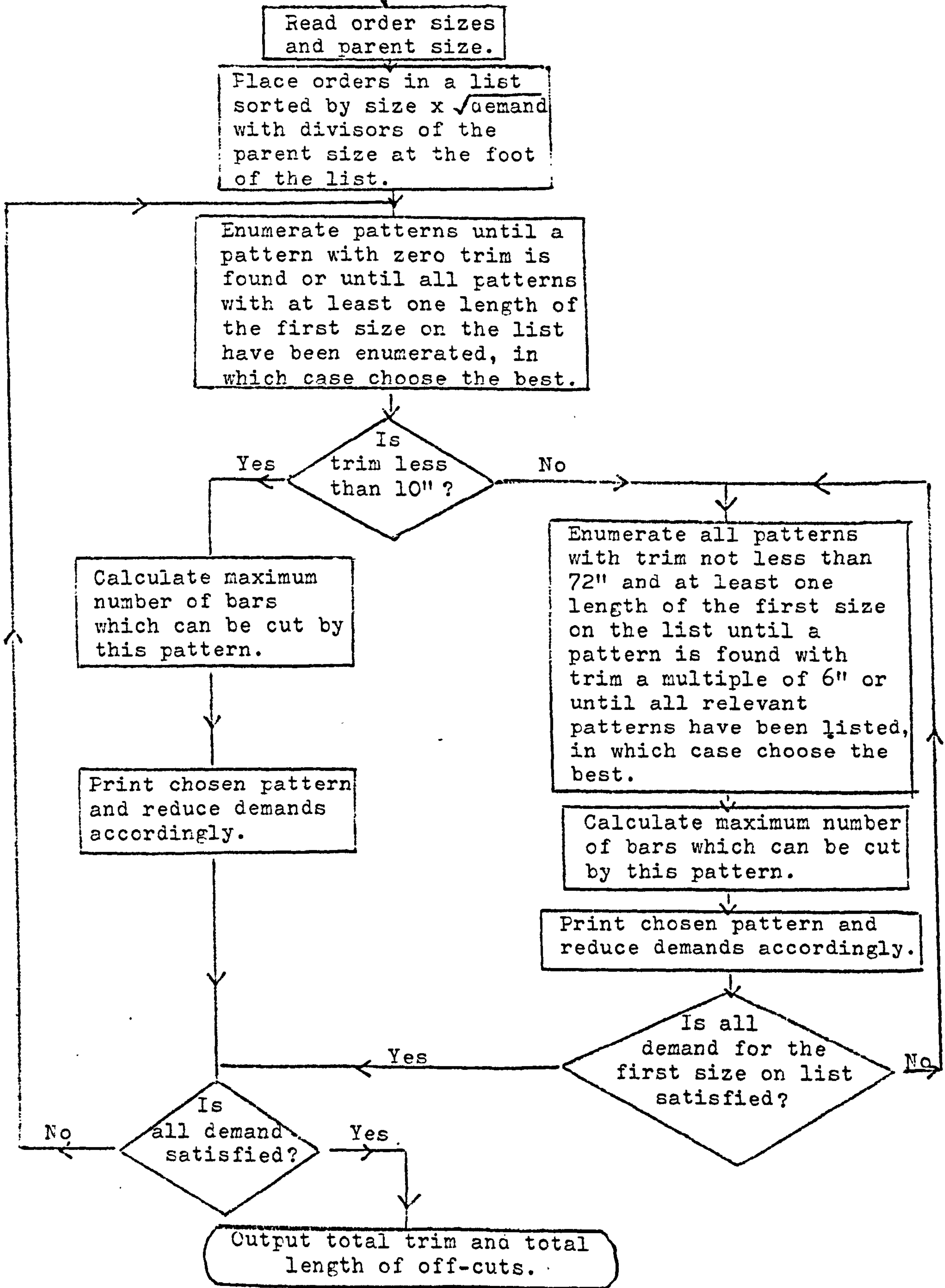


Figure 17. Flow diagram of the final pattern enumeration method for the bar cutting problem.

6.7 Subsequent Experiments

Two further experiments were performed using the pattern enumeration method developed in §6.6. In both experiments a full week's data of 402 orders was scheduled, rather than the seventeen orders used to develop the method.

The first experiment had two purposes. First, to show whether the method can easily be extended to handle a non-zero initial stock position, i.e. to consider patterns using off-cuts from earlier orders as well as parent sizes. Generally, longer lengths can be used to cut a wider variety of patterns and hence the chance of finding a suitable pattern is higher. Thus if patterns from off-cuts are considered on the same basis as those from parent lengths then those from parent lengths will tend to be chosen. This will lead to an unacceptable rise in the amount of off-cut stock to be held. The method of Program 11 (Appendix 3.2, p.148) outlined in Fig.18 therefore considers off-cut patterns before going on to those from parent lengths.

Secondly, in the initial experiments the pattern enumeration method had given rise to shorter off-cuts than the company's current method. It could be that these shorter lengths would accumulate in stock. It was thus necessary to run a full week's orders and examine whether the shorter off-cuts were used. The results of the experiment showed that the method of Fig. 18 handled the off-cuts successfully. The effect of the new off-cuts being of shorter length was balanced by fewer off-cuts being created.

The stock position remained reasonably constant throughout the week. Unfortunately as a result of the way the firm's data is handled it was not possible to make a direct comparison between these results and those which would have been obtained if the same 402 sets of orders had been scheduled by Stainton's method as currently used by the firm. However, over the 402 orders the average trim loss was 0.57%, roughly half that achieved by the company at present. The computation of the week's schedules took 810 computer secs. i.e. an average of 2 secs. per order, not an excessive amount of computer time.

The second experiment was designed to examine the effect on trim of the company's insistence that each of the orders was to be scheduled individually. The 402 orders scheduled in the first experiment were grouped by material and bar diameter and then all orders for

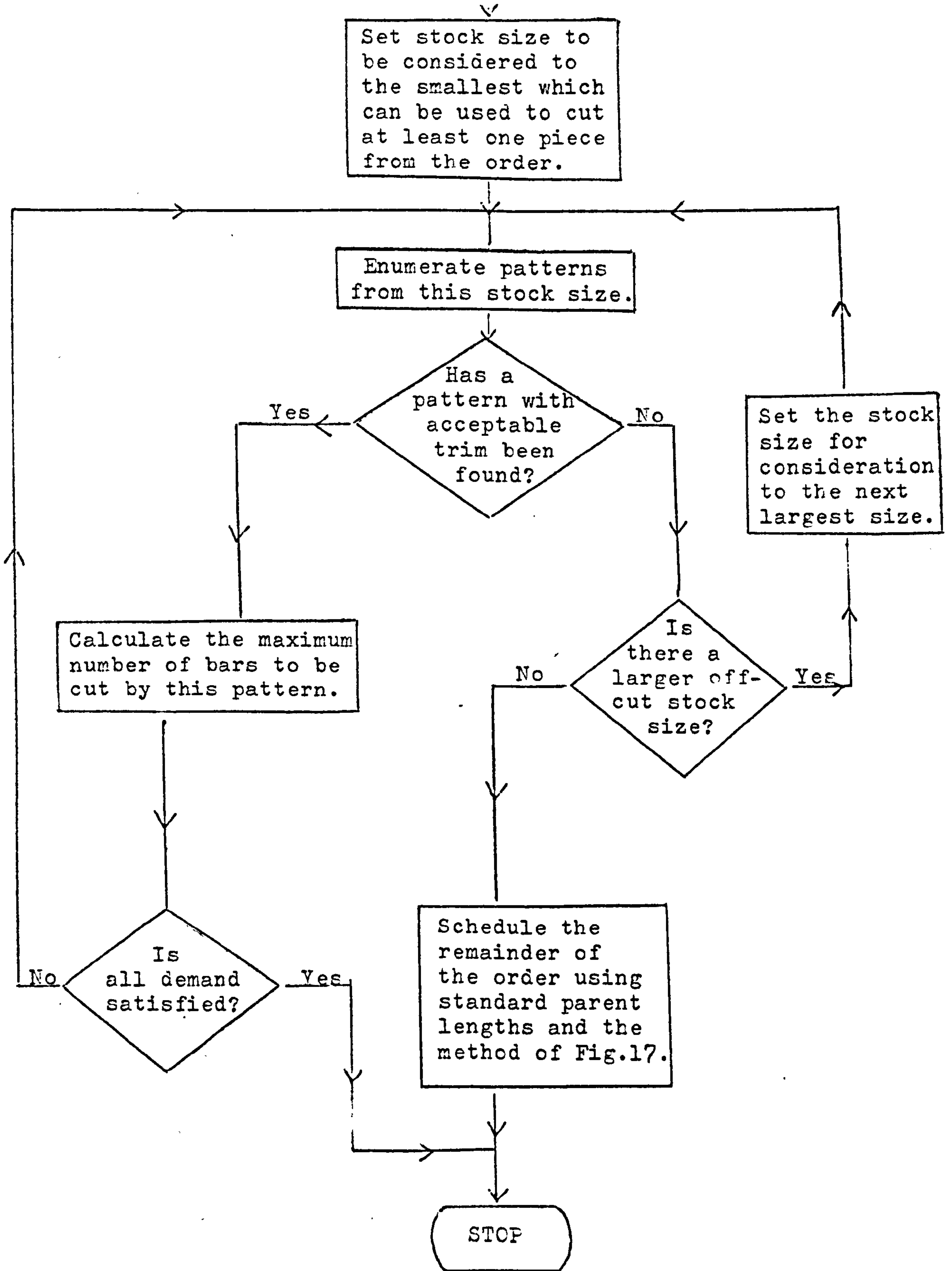


Figure 18. Flow diagram of the pattern enumeration method as adapted to consider off-cuts held in stock.

each of the seven types of bar were scheduled together. This not only reduced trim loss from 0.57% to 0.17%, but also reduced the amount of off-cut stock. The improvement is a result of having a wider range of patterns to explore and fewer intractable residual problems arising. When the firm carried out a similar experiment using Stainton's method the improvement gained by grouping orders was only slight. The pattern enumeration method seems more able to take advantage of the wider range of patterns than does Stainton's method.

As might be expected, searching through the wider range of patterns increased the computer time required to schedule the orders by a factor of three. Despite this increase in computer time the size of the reduction in trim must bring into question the wisdom of scheduling each order separately. It may be that much of the benefit in scheduling efficiency could be achieved by merely combining selected orders which individually would generate much trim. Alternatively it may be sufficient to allow the tail of one order to overlap the start of the next, thus avoiding many intractable residual problems. Either of these methods stands a good chance of reducing trim without incurring excessive material handling charges. If the pattern enumeration method were to be adopted by the firm, then this is one question which would have to be resolved so that the best overall scheduling system could be chosen.

6.8 Discussion

The scheduling problem presented in this chapter could, in theory, be solved by structuring it as an integer programming problem. While the solution obtained from such a procedure would minimize the cost of trim and off-cut stocks, it would be far from optimal overall as the computer costs involved in generating such schedules would outweigh the advantages gained. Indeed the dimensions of the problem are such that even approximations enabling linear programming methods of solution are very expensive in computer time. Furthermore the essentially integer nature of the problem requires ad hoc adjustments to be made to the linear solution to satisfy the non-linear restrictions. These ad hoc solutions move the schedules from optimality, and may yield solutions which are no better than those obtained using inexpensive heuristic techniques from the start.

As with the case study of chapter 5, it proved relatively easy to build up a pattern enumeration method which was adapted to the characteristics of the problem. The techniques developed earlier again proved sufficient to tailor the schedules to those required. Computer times of two seconds per order were achieved. The cost of such computer time is negligible compared to even small changes in the amount of trim.

In comparison with the firm's current scheduling method, the pattern enumeration method reduced trim loss in seventeen test orders to 19% of its previous value without any increase in the stock of off-cuts. It is unlikely that a reduction of quite this size could be achieved regularly in practice, as the method was developed around this data. However in the light of the subsequent experiment on 402 orders, a halving of trim loss would seem to be well within the bounds of the method. This improvement can be explained by the fact that Stainton's method tends to restrict consideration of patterns to those containing only one or two different order sizes. The pattern enumeration method, on the other hand, considers a far wider pattern set. This illustrates the importance of the procedure used to generate patterns for consideration by a heuristic method. The performance of the pattern enumeration method could be further improved by considering an even wider range of patterns if the need to schedule orders individually could be removed.

7. A SEQUENTIAL BAR CUTTING PROBLEM

7.1 The Scheduling Problem

In the case studies of chapters 5 and 6, as in most trim loss problems described in the literature, the size of the parent lengths - from which orders are to be cut - are known before scheduling begins. In these studies there may be uncertainty about future orders, requiring a flexible stock to be held (as in the previous case study), but not about the size of parent lengths. Parent lengths can normally be ordered in pre-set sizes, in the quantities required, from outside companies. Alternatively they may be manufactured within the company in the quantities and sizes required.

The case study presented here, however, does not conform to this general format. As a consequence of the technology involved, the size of parent lengths are not known before scheduling begins. Even after scheduling has started, only the size of the current parent length is available to the scheduler, but to balance this he has access to a relatively large order book. The uncertainty in future orders is therefore less significant. The problem to be solved is, in this respect, an inversion of the standard form of the trim-loss problem.*

Since the size of parent lengths only become known singly, the scheduling has to be carried out at the same time as the cutting operation. If cutting is not to be held up, the scheduling must be done quickly.

The company encountering these problems is the British Steel Corporation. BSC makes sectional girders for the construction industry at its Teesside works. These girders are manufactured by rolling ingots of hot metal until they reach the required cross-section and gauge. Each rolling increases the length of the girder and thus it is not until the rolling is completed that the length of the finished girder is known. The girder is then cut into order lengths while still hot. While it is being cut the next ingot is being rolled. The production sequence for the girders is as shown in Fig. 19.

* The only other paper which appears to tackle a problem of the type considered here (Tilanus & Gerhardt (20)) is, as in this case, concerned with the steel industry, but other characteristics differ.

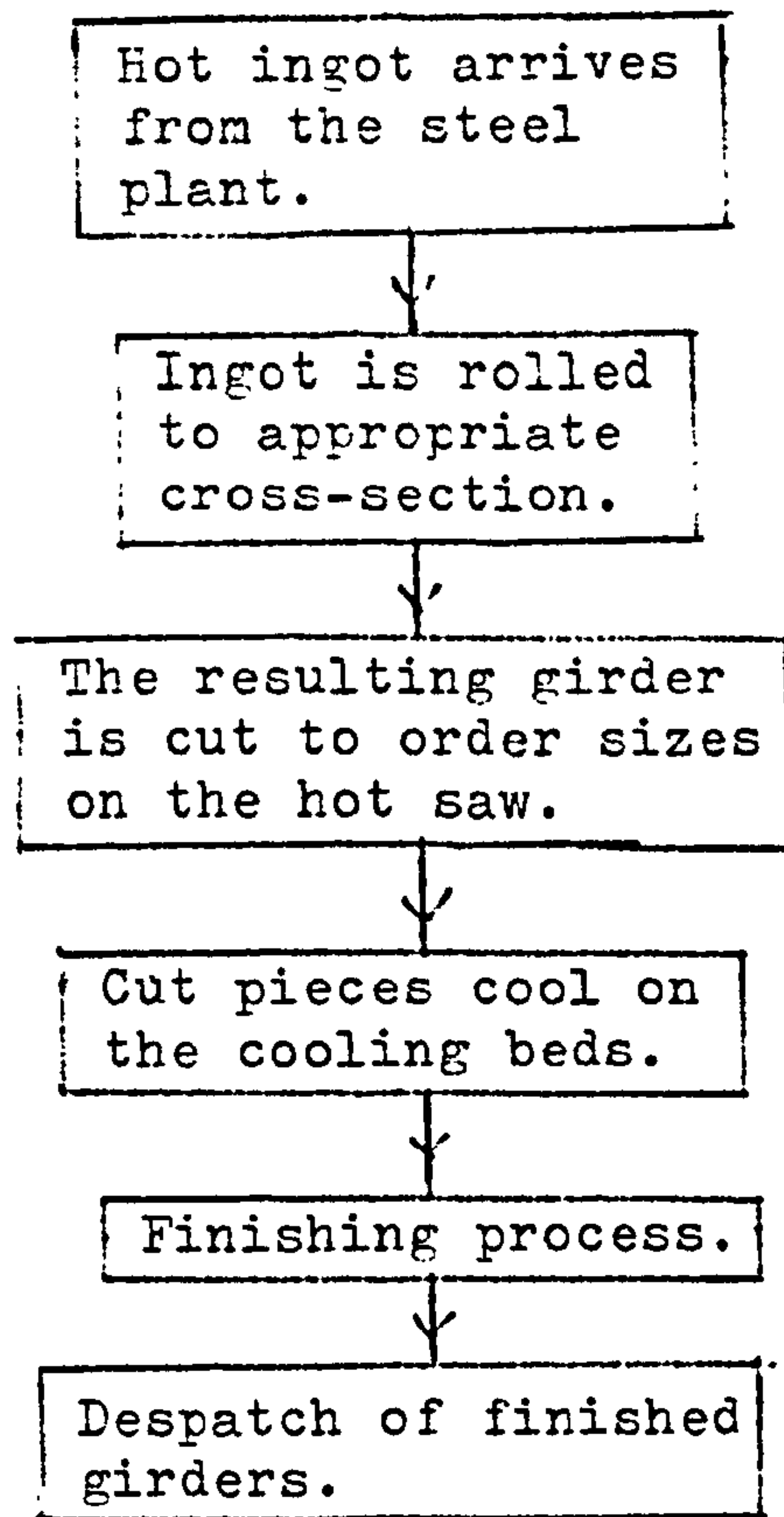


Figure 19. Production sequence for steel girders.

The ingots are rolled into many different cross-sections and gauges. As in the previous case study, each different cross-section and gauge can, for scheduling purposes, be viewed as a different product. Orders for a particular cross-section and gauge are collected and produced as a group.

The process used to cut the hot girder is to run the girder up to the hot saw so that the crop end may be trimmed off (each end of the parent girder is slightly mis-formed and can not be used); the order sizes required are then cut one at a time by lining up the appropriate point in the parent girder with the hot saw and cutting. As this lining up process must be done for each girder irrespective of whether the pattern has changed, set-up costs may be disregarded.

Production, as in the previous case study, is to meet the current order book with one exception. Occasional off-cuts can be stocked if the only alternative is incurring very heavy trim loss.

The key characteristic of this problem is the reduction of trim under the uncertainty concerning the lengths of future parent girders. As the output of girders was 410 000 tonnes/year at the time of the study, even small percentage trim savings can be of considerable economic benefit.

7.2 Detailed Characteristics

The average parent lengths after rolling for different types of girder range from 50m. to 90m. For girders of a particular cross-section and gauge the length can vary by up to 50m. The distribution of parent lengths for the first test problem (data set 39, appendix 4.1 p. 158) is shown in Fig. 20.

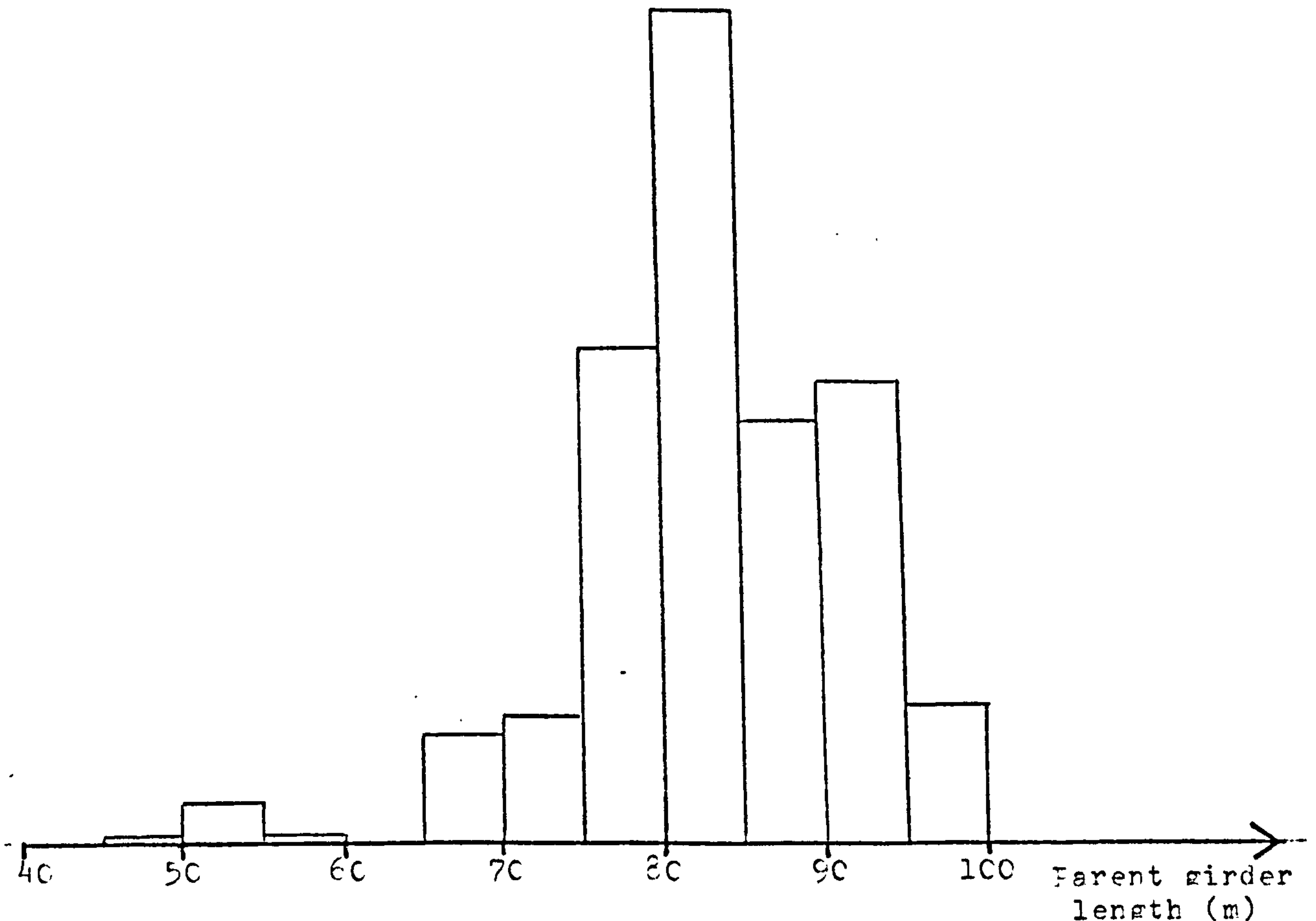


Figure 20. Distribution of one set of parent lengths.

Orders are stipulated to the nearest millimetre but mill practice, as a result of limitations in the accuracy of the cutting method, is that, for scheduling purposes, lengths will be rounded down to the nearest centimetre. The data (Appendix 4.1, p. 158) for this study has similarly been rounded down to the nearest centimetre.

The order sizes range from 1m. to 20m. Usually about seven finished lengths can be cut from each parent girder. As a result, there are a reasonably large number of patterns to be considered at each stage.

The conditions under which an off-cut girder will be sent to stock are similar to those in the previous case study. There are no fixed stock sizes but whenever a combination of orders will generate

an unacceptable trim, several ordered pieces will be cut from a parent length and the remainder sent to stock. The pattern chosen will be such that the length sent to stock will not normally be less than 9m. Off-cuts of less than 9m. would be unlikely to be used later. Even with off-cuts of longer than 9m. not all stock will be used. Some will eventually be scrapped. The amount scrapped varies with market conditions. A recent study (7) estimates the percentage of off-cut material which will be scrapped as 40%. This figure has been used throughout this case study as a guide to total scheduling losses, i.e.

total amount scrapped = trim + 0.4 x amount sent to stock.

As market conditions change one would like to modify the balance between trim and off-cuts. This can be achieved by adjusting the acceptable trim parameter in the pattern enumeration method developed below.

Once orders are cut they are dispatched to customers direct from the cooling banks. It is therefore convenient to keep all lengths of one cross-section and gauge for a particular customer together. As there are only a limited number of cooling banks and only one customer's order can be kept on a cooling bank at once, a restriction is imposed that a maximum of six orders be open simultaneously. This restriction simplifies the problem of selecting a suitable pattern by reducing the number of patterns to be searched, but introduces the problem of how to select orders to be opened.

7.3 Current Scheduling Method

At the time of the study, scheduling was being done manually by the operator of the hot saw as the girders arrived from rolling to be cut. The British Steel Corporation are, however, considering using a computerised scheduling procedure to improve scheduling performance. Work has been done by Evans and Quarrington [(7) & (8)] to model the behaviour of the manual scheduler and to improve upon it. There are thus, as in the case study of chapter 5, two methods of solution which can be used for comparison with the pattern enumeration technique, i.e. the current manual method and the Corporation's proposed computer method.

The manual method is constrained by the number of patterns which can be considered before each cut is made. The sawman has apparently built his technique so as to give decisions quickly rather than to

produce the most efficient schedules. Before starting to cut a set of orders he divides them into 'main' and 'saver' types. 'Main' orders consist of large quantities of a few different lengths and are thus difficult to schedule efficiently. 'Saver' orders contain a larger variety of lengths with only a few pieces of each length required. For each parent length the sawman will aim to use sizes from one 'main' order and one 'saver'. As a parent length approaches the saw he first completes the cutting of any partially completed orders with only a few lengths remaining. This enables the cooling beds to be kept reasonably clear. He then finds the length in the current 'main' order with the largest remaining requirement. Lengths of this size will then be cut until only about 30m. of the parent girder remains. If the 'main' order is completed during this cutting, a new one will be opened which will give a low trim. Once the 30m. point is reached the sawman selects a good combination of 'main' and 'saver' order lengths to achieve a trim of less than 75cm. If this is not possible then he will consider opening a new 'saver' order or creating an off-cut. This process is summarised in Fig. 21.

The computer method [(7) and (8)] proposed by BSC to replace the manual method, described above, is a heuristic procedure designed for this particular problem. It is able to consider a wider range of patterns than is possible manually, enabling savings in trim and off-cuts to be made. Unlike the manual method, a complete pattern is chosen before cutting a girder is started.

At present the sawman is unable to see the end of the parent girder until there is only 40m. or 50m. remaining. It is unlikely, however, that he would be able to improve his performance even if this restriction were not present, as he barely has time to consider the restricted set of patterns for the last 30m. A computer, on the other hand, would be able to take advantage of the earlier information regarding girder length. Sanders (18) suggests one way that this information can be collected by the use of television cameras. The ESC computer method assumes that, using a method such as that of Sanders, the parent girder length is known before scheduling starts.

With this extra information a reduction in scrap material from 1.35% for manual scheduling to 1.24% has been achieved by ESC's computer method (8) on five test problems (Tables 33 and 34 Appendix 4.3, p. 181 show these results in more detail). Further work

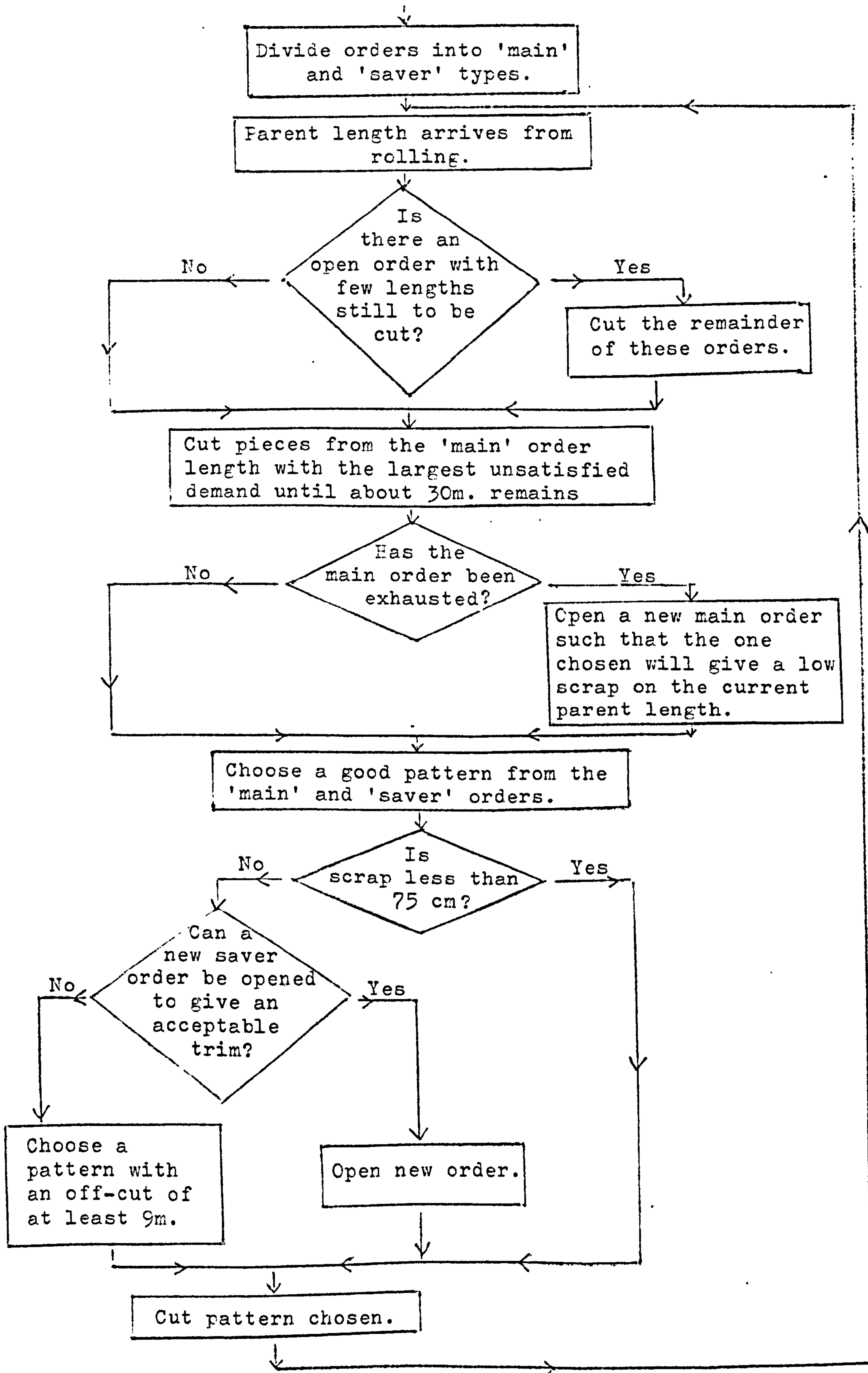


Figure 21. Flow diagram of the manual scheduling method.

done by British Steel suggests that the total scrap can be reduced to about 0.8% by using some of the techniques developed for pattern enumeration below. The improvement over the manual method is, presumably, a result of being able to consider each parent girder as a whole, and also by considering patterns in a more systematic manner.

7.4 Proposed Method

This problem can not easily be formulated for solution by either linear programming or branch and bound. This is largely due to the fact that not all the parent sizes are known at the start of the scheduling procedure and thus scheduling must be done sequentially. Even if a modification of a standard method were used, it would be necessary at each stage to place some value on the state of the order book after the current parent length had been cut (i.e. how easily the order book can be scheduled); this could only be done heuristically as future parent lengths are not known. If heuristics are to be used as a part of a linear programming or branch and bound method then the advantage (i.e. of optimality) of these methods is lost. Section 6.4 has shown the dangers involved in modifying LP solutions. Therefore, the natural approach to this problem is a heuristic one, such as that taken by British Steel or that of pattern enumeration.

The development of a heuristic method based on pattern enumeration will now be considered. Since the main difference between this case-study and earlier ones is that the scheduling process is sequential, (i.e. not until the current parent length has been 'cut' can the next length be made known to the scheduling procedure) the aim at each stage of the procedure must be not only to generate an efficient pattern for the current parent length, but also to make as wide a choice of patterns available to the next stage of the scheduling procedure as possible. This will reduce the risk of high scrap patterns having to be chosen later. Three main elements within a pattern enumeration method will determine the pattern chosen at each stage and the nature of the residual problem.

First, the rule used to determine when to open a new order and which order to open, will affect not only the range of order lengths and hence patterns to be considered at each stage, but also the residual problem. Had there not been a restricted number of cooling banks then maximum flexibility could have been achieved by having all the orders open at once. This would have enabled a very wide range of

patterns to be available at each stage. However given that there is only room on the cooling banks for six orders at any one time, there is a conflict between the two aims. To obtain the greatest choice of patterns at any stage one would like to have as many orders open as possible, i.e. six. On the other hand this restricts the choice available when scheduling the next parent length as no new orders can be opened no matter how low a trim they would have yielded. The best situation, after cutting the current girder, is to have as few orders open as possible. New orders can then be opened, as required, once the size of the next parent girder is known. A pattern enumeration method is developed, therefore, in which new orders are only opened when none of the patterns using currently open orders yields an acceptable trim. Whenever a new order is to be opened each of the unopened orders is examined in turn until one is found which, either on its own, or in combination with the currently opened orders, yields acceptable trim. This order is then used. This method chooses only the first acceptable order encountered and does not compare all the different orders for a best solution. The sequencing of orders is therefore important. Initially the sequence in the order book was used. The effect of different order sequences will be examined later.

Secondly, the sequence in which open order lengths are listed will affect the pattern chosen at each stage, lengths at the head of the list being more likely to be used. At first sight the best listing is that in which lengths from the same order are kept together and these groups of lengths are listed by the total demand in the group still unsatisfied. There would then be a strong tendency to choose lengths from the first group, enabling that order to be met completely and hence a cooling bank to be freed. Such a procedure would, however, have one big draw-back in that orders containing large demands for one or two lengths would remain permanently at the foot of the list, while orders only recently opened would come in above them. The final result of this would be that there would be a very limited range of demand lengths (all with quite a large quantity required) left towards the end of the scheduling procedure which would inevitably lead to several high trim patterns having to be used. A more useful ordering would thus be to introduce newly opened orders to the foot of the list of order lengths for enumeration. This procedure was adopted.

Thirdly, the pattern acceptance criterion will affect the nature of the schedules generated. If a very demanding acceptance criterion is used (i.e. very low trim) then it is more likely that extra orders will have to be opened before a pattern satisfying the criterion is found. Although a series of girders will be cut with very low trim, the cooling banks will eventually become blocked and a pattern incurring high trim will have to be cut, cancelling out previous gains. On the other hand if the acceptance criterion is too weak, then a series of mediocre patterns will be used, but the cooling banks will not become blocked. A balance will have to be struck. Initially in this method the acceptance criteria were set arbitrarily to 20cm. trim for patterns using only currently open order lengths and 10cm. for patterns requiring a new order to be opened. (This was to ensure that an order was not opened unnecessarily.) If during enumeration a pattern with a trim of less than half that acceptable is found then enumeration will stop at that point and the pattern be used. When no new order meets the acceptance criteria then the best pattern from the current orders will be used provided that it has a trim less than 4m., otherwise a pattern sending an off-cut to stock will be used. This parameter of 4m. can be decreased to reduce trim and increase the quantity of off-cuts (or increased to have the opposite effect) to reflect the change in demand for material from stock as market conditions vary. The setting used reflects practice in the mill at the time of the study.

When developing the method it was necessary to consider not only the efficiency of the schedules generated, but also the time taken to produce them. Whereas in the earlier case studies computing time was only a small cost to be set against the savings achieved, here it is a major constraint, there being only a short time between the length of a parent girder being known and the first cut having to be made. The pattern enumeration method of Fig. 22 has been developed with this consideration in mind; for example, no attempt is made to find the best order to open at any stage, but merely to find one which can yield a pattern meeting the acceptance criteria. Even so, the number of patterns to be considered is quite large. Methods of improving computational efficiency will have to be examined. In particular some of those of §4.5, p.40 are considered relevant and are tested below.

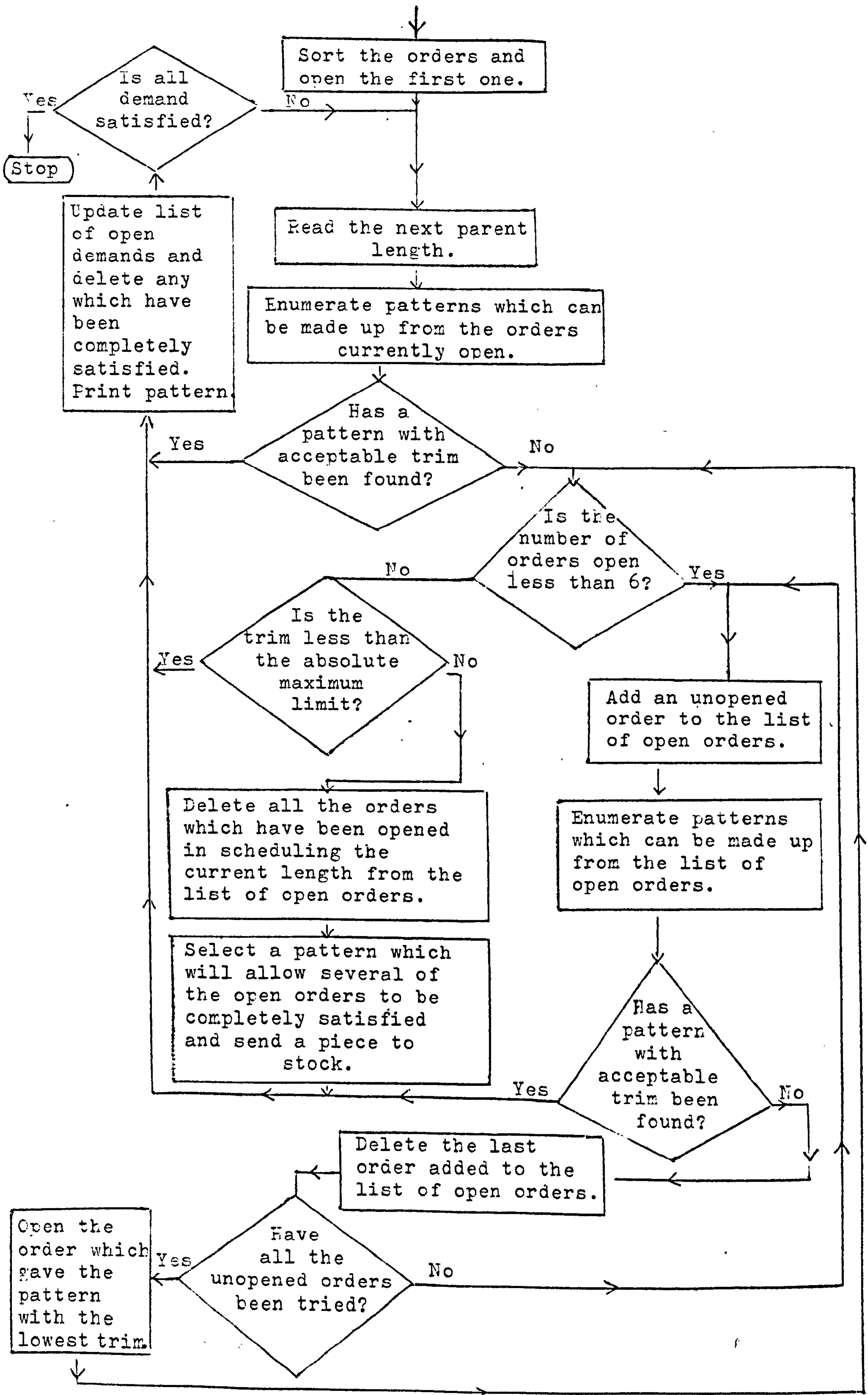


Figure 22. Flow diagram of the pattern enumeration method.

For ease of comparison with the British Steel computer method, the pattern enumeration method has been developed on five sets of data (data sets 39-43, Appendix 4.1, p.158), the parameters tuned on twenty sets of data (data sets 39-58) and finally tested on the twenty sets of data but with different parent lengths. The process follows exactly that taken by Evans [(7) and (8)] in developing British Steel's computer-based method.

Table 10. Summary of the schedules from the first pattern enumeration method.

	Manual method	British Steel's Heuristic	Pattern Enumeration
Number of girders used	197	197	198
Trim (%)	0.71	0.58	0.31
Off-cut (%)	1.59	1.66	2.60
Estimate of total scrap (%)	1.35	1.24	1.35

The results of the first set of schedules are given in Tables 33, 34 and 35 in Appendix 4.3, p.181 and are summarized in Table 10 above. The performance of the pattern enumeration method is comparable with the manual method, but is inferior to British Steel's heuristic. The pattern enumeration method has produced an excessive amount of off-cuts. These had been produced towards the end of the scheduling procedure when only one or two orders still remained to be satisfied. In the scheduling of data set 39, for example, the last four girders had to be cut with only one order remaining open. This order contained a large number of pieces of only one length, restricting the choice of patterns severely.

Scheduling efficiency can be improved by adjusting the three key characteristics discussed above. Turning to the first of these characteristics (the selection of orders to be opened): changing the order sequence will leave a different residual problem to be solved towards the end of the scheduling process. If residual problems involving large quantities of a few lengths are to be avoided, then these orders will have to be scheduled early by placing them at the top of the order list. The program was altered so that the orders were sorted by the total number of pieces required by the order. With this alteration the scrap percentage was reduced from 1.35% to 0.78%, mainly by a reduction in the amount of off-cuts. A parent girder was saved in the scheduling of data sets 41 and 43 (more

detailed results are shown in Table 36 , Appendix 4.3, p.182). The alteration reduced wastage from the last few girders, thereby significantly improving the quality of the schedule.

Despite these improvements there was a tendency for few orders to be open at the start of a schedule while the maximum number of six was open towards the end. This was particularly noticeable in data sets 41 and 43, and was caused by orders requiring only one piece to be cut being grouped together at the end of the demand list. Leaving these orders to the end of the schedule meant that six orders (instead of the usual two or three) had to be open simultaneously to provide a satisfactory range of lengths for pattern building. Thus the ordering method needs to be changed to allow for two things:

- (a) The average quantity required per length in the order;
- (b) The spread of lengths within an order.

These two factors were combined together in an ordering method developed by Evans and Quarrington for British Steel (8) namely to sort the orders by :-

Total number of pieces required by the order

$$\left(\begin{array}{l} \text{Number of different lengths} \\ \text{within the order} \end{array} \right) \times (1 + \text{Max.length} - \text{Min.length})$$

This order sequence was tested within the pattern enumeration method.

Table 11. Summary of the schedules yielded by different order sequences.

	Manual method	British Steel's heuristic	Pattern enumeration		
			Unsorted	Sort by number of pieces	Sort using Evans & Quarrington criterion
Number of girders used.	197	197	198	196	196
Trim (%)	.71	.58	.31	.21	.20
Off-cut (%)	1.59	1.66	2.60	1.45	1.45
Estimate of total scrap (%)	1.35	1.24	1.35	.78	.78

While this change did not reduce the percentage scrap it did make the number of open orders far more stable. It has therefore been used in all later runs as the alternative (which frequently requires six orders to be open towards the end of the schedule) will eventually yield an intractable residual problem. (With six orders open there is

a likelihood of being forced to use a high trim pattern, particularly if each order contains only one length, as no new orders may be opened to increase the number of feasible patterns). While further experiments on the order sequence could possibly improve the schedules marginally, the Evans and Quarrington's sequence had no obvious faults. Attention was therefore directed to other important determinants of the pattern enumeration method's success.

Grouping the order lengths from each open order together seemed to be working reasonably satisfactorily. Other than orders for very large amounts, orders were being completed quite quickly, freeing the cooling banks. However some orders for a very large number of pieces of a single length remained open right through the scheduling procedure creating awkward residual problems. This could be resolved by relaxing the pattern acceptance criteria, or by allowing these orders to be handled differently from the rest. The possibility of a special routine will be re-examined after modifying the acceptable trim parameters.

Initially the acceptable trim parameters were set to 20cm. for patterns using currently open orders and 10cm. for those requiring a new order to be opened. These parameters were used irrespective of the number of orders open. However, it is less restricting to move from, say, two orders open to three than from five to six (in which case no further orders may be opened). If the acceptable trim parameters are made dependent on the number of orders open, then control over the method is enhanced. One can insist on very low trim when the loss of flexibility in opening a new order is small and accept a higher trim when the loss of flexibility would be great. The acceptable trim parameters for 0, 1, 2, ..., 6 orders open were set respectively at 0cm, 0cm, 5cm, 5cm, 10cm, 10cm, and 15cm. The method was then used to schedule the full twenty sets of data. The results are shown in Tables 38 and 39 Appendix 4.3, p. 183 and summarized in Table 12 below.

Table 12. Summary of the schedules yielded when the acceptable trim parameters change with the number of orders open.

	British Steel's heuristic	Pattern enumeration
Number of girders used	1 451	1 445
Trim (%)	.65	.20
Off-cut (%)	1.48	1.63
Estimate of total scrap (%)	1.24	.86

Despite the improved control over pattern acceptance given by varying the acceptable trim parameter, orders for a large number of pieces of one length were still leading to excessive wastage. For example, one order in data set 48 requires 200 pieces of one length to be cut. This length was being cut long after all the other orders had been completed. The pattern enumeration method used three more girders to schedule this data set than did British Steel's method. British Steel had achieved their improved performance by forcing lengths required in such large quantities into early patterns, even if it increased trim. A similar procedure will have to be adopted within the pattern enumeration method. Whenever the number of pieces to be cut of one length is greater than three times the number of unopened orders, then a pattern will be cut using as many as possible of the length with large demand.

From a detailed examination of the schedules summarized in Table 12 it was seen that several required six orders to be open for most of the schedule. This indicated that the acceptance criteria had been set too tightly, forcing extra orders to be opened. The acceptable trim parameters were therefore increased to 0cm, 0cm, 5cm, 10cm, 15cm, 20cm and 25cm. A new set of schedules were produced using these parameters and the special routine for large quantities described above. The results are summarized in Table 13 below. (More detailed results are given in Table 40, Appendix 4.3, p.184).

Table 13. Summary of the schedules obtained when orders for large quantities are forced into solution.

	British Steel's heuristic	Pattern enumeration	
		Large orders not forced into solution	Large orders forced into solution and relaxed trim criteria
Number of girders	1 451	1 445	1 439
Trim (%)	.65	.20	.20
Off-cut (%)	1.48	1.63	1.14
Estimate of total scrap (%)	1.24	.86	.65

The modifications made have improved the quality of the schedules mainly by reducing the risk of intractable residual problems arising. In particular, the data set involving a high demand for one length (data set 48) has been scheduled using fewer parent girders than before.

However the rule used to force a particular order length into solution resulted in the normal pattern enumeration being over-riden more often than was necessary, i.e. lengths were forced into solution even when they would not have yielded intractable residual problems. The rule was therefore weakened so that the special routine was only entered if the unsatisfied demand for a length was greater than four times the number of unopened orders plus twice the number of lengths currently available for scheduling (this criteria was arrived at by a subjective evaluation of the results of the previous runs). The program with these modifications is shown as Program 12, Appendix 4.2, p.175. The consequences of this modification are shown in Table 41, Appendix 4.3, p.184 and are summarized in Table 14 below.

Table 14. Summary of the schedules obtained with a modified rule for forcing large orders into solution.

	British Steel's heuristic	Pattern enumeration	
		Original decision rule	Modified decision rule
Number of girders	1 451	1 439	1 440
Trim (%)	.65	.20	.18
Off-cut (%)	1.48	1.14	1.21
Estimate of total scrap (%)	1.24	.65	.66

The modification reduced the amount of trim produced since fewer patterns were generated by the special routine (which did not require patterns to meet the acceptance criteria). The increased amount of off-cuts due to an extra girder being required to cut data set 45, was purely a result of the particular lengths left in the residual problem. To amend the parameters for this particular data set could simply throw up similar awkward residual problems on other sets. On this occasion the special routine had only been entered when necessary and so the procedure was not changed.

After the pattern enumeration method was developed for this case study, Evans modified the British Steel heuristic. Instead of keeping a fixed number of orders open, orders are opened only when it is necessary to reduce trim, (i.e. a similar rule to that used within the pattern enumeration method). Table 15 shows that this improved the performance of his heuristic considerably. (More detailed results are given in Table 42, Appendix 4.3, p.185). This improvement does much to close the gap between the British Steel heuristic and the pattern enumeration method.

Table 15. Summary of the schedules obtained by the modified British Steel heuristic.

	Original British Steel heuristic	Modified British Steel heuristic	Pattern enumeration
Number of girders	1 451	1 444	1 440
Trim (%)	.65	.30	.18
Off-cut (%)	1.48	1.36	1.21
Estimate of total scrap (%)	1.24	.84	.66

Because of the nature of the production process, the time taken to generate schedules is important in this case study. Techniques to reduce computer time will not only reduce computer expenditure, but also determine the viability of any scheduling method. The computational techniques of §4.5, p.40 will have to be examined to see if any method can be used to improve computational efficiency. Examination of the order lengths to be scheduled shows that many (but not all) are in multiples of 10cm, 20cm or 25cm. The technique of using the highest common factors of the order lengths to skip poor patterns may, in this situation, be able to reduce computer times. Without the use of highest common factors an ICL 1905E took 2 182 seconds to schedule the 20 problems discussed above, i.e. approximately $1\frac{1}{2}$ seconds/girder, whilst using highest common factors this dropped to only 383 seconds, i.e. $\frac{1}{4}$ second/girder. This should give a virtually instantaneous response to the sawman.

To test the effectiveness of the completed pattern enumeration method rigorously it should really be used to schedule a completely different set of problems from that on which it has been developed. Unfortunately a separate set of problems with solutions by the British Steel heuristic (as a base for comparison) was not available. Instead the method was tested on the same set of orders but with different parent lengths. Even so, results were only obtainable from British Steel for their original heuristic. Comparison between the two methods is therefore somewhat unfair as the final pattern enumeration method is being compared with an intermediate British Steel method. The stability of each method to changes in the parent lengths can, however, be examined. Table 16 shows that neither method is unduly affected by the change in parent lengths. The pattern enumeration method is particularly stable because, as explained above, parameters were chosen with a view to stability rather than achieving the best

possible schedules with the particular lengths in the first set of parent girders. (More detailed results are given in Tables 43 and 44, Appendix 4.3, p. 185 & 186).

Table 16. Summary of the schedules obtained using the second set of parent lengths.

	British Steel heuristic		Pattern enumeration method	
	1st set of parent lengths	2nd set of parent lengths	1st set of parent lengths	2nd set of parent lengths
Number of girders	1 451	1 456	1 440	1 441
Trim (%)	.65	.61	.18	.20
Off-cut (%)	1.48	1.90	1.21	1.15
Estimate of total scrap (%)	1.24	1.37	.66	.66

7.5 Discussion

This case study concerns a trim loss problem in which the standard characteristics of trim loss problems are inverted. Normally, there is little uncertainty about the supply of parent lengths but the future demand for different order sizes is unknown. Here the opposite is true. Only the size of the current parent length is known to the scheduler while, as a result of a large order book available, there is relatively little uncertainty about demand. Despite these differences it was possible to develop a successful pattern enumeration method along the lines used previously. Success again depended on the avoidance of intractable residual problems by ensuring that flexibility was maintained and that potentially awkward order lengths were scheduled early in the procedure. This was achieved by suitable choice of pattern acceptance criteria and manipulation of the sequence in which patterns are enumerated.

It is interesting to note that Evans was able to improve his heuristic substantially, reducing overall scrap from 1.24% to 0.84% by only opening orders as required. This is additional evidence in support of one of the main contentions of this thesis, that an effective scheduling procedure can be built around a simple pattern generation procedure provided that care is taken to tailor the pattern generated at each stage to the type required. In particular, Evan's modification changed the list of order lengths from which a pattern will be chosen. The list of order lengths to be used in a pattern has

emerged in all three case studies as one of the most important determinants of the success of a pattern enumeration method.

The pattern enumeration method developed here and the final British Steel method are very similar, as one might expect when each was aware of the other's work. Both are heuristic methods aiming to produce satisfactory, but not necessarily optimal, schedules. The decision rules used to open new orders are similar, and the order sequence of Evans and Quarrington was adopted in the pattern enumeration method. The main difference lies in the technique used to generate patterns for consideration. Pattern enumeration is able to generate feasible patterns simply and systematically. Good patterns are not missed. This presumably accounts for most of the remaining difference between the two methods; an overall scrap of 0.84% from British Steel's heuristic and 0.66% from pattern enumeration.

The development of the pattern enumeration method again showed the importance of certain of the techniques to improve computational efficiency. When the highest common factor technique was used, the computer time to generate all twenty schedules dropped from 36 minutes to 6 minutes.

8. CONCLUSIONS

8.1 Discussion of the Case Studies

Most of the distinguishing characteristics of practical trim loss problems (in one dimension) are found in one or other of the case studies of chapters 5, 6 and 7. The case study of chapter 5 is typical of the scheduling problems of the paper industry. Rolls have to be slit to meet customer requirements. Because paper is relatively inexpensive, production costs are not dominated by trim-loss costs, other costs (particularly set-up costs) are equally important. The case study of chapter 6 is a simple bar cropping problem where the aim is to minimize the amount of trim and off-cuts. In this case set-up costs are unimportant. The problem in chapter 7 is again quite different; its key characteristic is the lack of knowledge about the size of future parent lengths. Not only do the problems differ in these respects but also, as Table 17 shows, in other features. Despite these differences it has proved possible to follow the same basic strategy in developing a pattern enumeration method for each of the problems.

Table 17. Comparison of the characteristics of the three case studies.

	Case study of chapter		
	5	6	7
<u>Orders</u>			
Average number of different sizes to be scheduled together.	20	13	100 (but not all these are 'live' at once)
Tolerance on order quantities.	Yes	No	No
<u>Parent Sizes</u>			
Number of parent sizes of each material.	1	1+ previous off-cuts	Many, but only the current one is known.
<u>Patterns</u>			
Average number of pieces in a pattern.	7	4	7
Off-cuts permitted.	No	Yes, but only in certain sizes.	Yes
Limit on the number of pieces in a pattern.	Yes	No	No
Significant set-up costs.	Yes	No	No
<u>Production Process</u>			
Material being processed.	Paper	Steel rods	Steel girders
Cutting method.	Slitting	Cropping	Sawing
Type of schedule required.	Integer	Integer	Integer

The development of a successful pattern enumeration method for each of the case studies depended on avoiding intractable residual problems. Excessive costs arose in the case study of chapter 5 when either narrow widths had to be cut or when short run lengths were used, in chapter 6 they arose when orders for large quantities of a few lengths remained unsatisfied towards the end of the scheduling process, and in chapter 7 when the cooling banks became blocked. In each situation, however, high costs were avoided by detecting those order sizes most likely to lead to such problems and ensuring that these were scheduled early. A similar process could have been used in developing methods around other heuristic techniques for generating patterns. The advantage of pattern enumeration is the ease with which this can be done.

Table 18. Performance of the pattern enumeration method compared to other methods.

	Case study of chapter		
	5 ¹	6 ²	7 ³
% reduction in costs relative to manual methods	30%	Manual schedules not available.	42%
% reduction in costs relative to current computer methods.	19%	81%	21%

Notes

1. Comparing cost in column 3 with that in columns 1 and 2 of Table 3, p. 58.
2. Comparing trim-loss in columns 1 and 3 of Table 9, p. 78.
3. Comparing total scrap in column I with that in columns A and J of Table 45, p. 187.

The pattern enumeration methods developed have yielded schedules which, when compared to either manual methods or other computer heuristics, have performed satisfactorily. Since none of the problems could be formulated conveniently for solution by an optimizing method (e.g. LP) it was not possible to assess how close to optimality the pattern enumeration schedules were. From a theoretical point of view, this is unfortunate as it makes it impossible to establish the effectiveness of these pattern enumeration methods against an absolute base-line. Practically, this is less of a problem. What is important is to establish a significant improvement over current solution methods for these case studies, bearing in mind that computing costs must not

outweigh the savings gained. In each of the case studies, computing costs were less than a tenth of the improvements in scheduling efficiency.

In each of the case studies, attention was directed to the efficient selection of cutting patterns. The restrictions in the problem definitions given by the firms in question were generally accepted. For example, in chapter 7 the limit of six on the number of cooling banks was used without scrutiny. It may well be that increasing the number of banks could reduce trim costs substantially by allowing more orders to be open at once. Running the model with more banks (and suitably modified parameters) would show whether such a move was economically worthwhile. Obviously in a thorough O.R. study of the scheduling problems facing each of the three firms these issues would have to be tackled. However, the subject of this thesis is the pattern selection problem, and so other issues have generally had to be laid aside. In the only case where a restriction was examined, (chapter 6) it was found to have had a significant effect on the ease with which the orders could be scheduled.

8.2 Evaluation of the pattern enumeration approach

Experience has shown that while pattern enumeration methods may be used in a range of situations, the actual methods will differ as the characteristics of the problem differ. This means that it will not be possible to produce a standard pattern enumeration package in the same way as, for example, standard LP packages are produced. Nevertheless the solution procedures for the three case studies have far more in common than merely the procedure used to generate patterns (i.e. pattern enumeration). The same steps have been taken in each study in developing an effective scheduling method.

In a heuristic method such as that proposed, in which patterns are accepted into solution one at a time, inefficient patterns (i.e. those with high trim, involving off-cuts or incurring reprocessing costs) tend to have to be used towards the end of the scheduling process. At this point most of the demand requirements have been satisfied and so only a restricted choice of demand sizes (and hence patterns) remain. With few patterns available it is unlikely that any of them are efficient. The success of a scheduling method therefore depends on its ability to avoid these intractable residual problems.

In each of the case studies a set of orders was first scheduled using a simple pattern enumeration method. These schedules were then examined to see the nature of the residual problems which arose. The pattern enumeration method was then modified so that the likelihood of intractable residual problems arising was reduced. In the case study of chapter 5 this was done by considering the nature of the residual problem which would occur if a particular pattern was cut, before deciding to accept it. In chapters 6 and 7 a far simpler procedure was adopted. Intractable residual problems could be associated with certain order sizes. It was thus possible, by placing these at the head of the list of sizes for enumeration, to ensure that they were scheduled early. Once the outline of the method was thus established the pattern acceptance criteria were set to give a reasonable balance between the different production costs; balancing, for example, lower trim against increased off-cuts.

These steps for developing an effective scheduling method could be followed with other techniques for generating patterns, e.g. Stainton's (19). We thus need to consider the relative advantages and disadvantages of pattern enumeration.

In §4.1, p.23, five criteria (besides scheduling efficiency) were outlined which needed to be satisfied before a method for solving trim loss problems could be said to be applicable to a wide class of one-dimensional trim loss problems. From the experience gained in the three case studies, the range of application of a pattern enumeration approach can be assessed using those criteria:

- (a) "It should be possible to include within the formulation of the method not only the constraint that demand be satisfied, but also the many other common constraints which arise in practice".

As Table 17, p. 102 shows, several extra constraints have been encountered in the case studies: a limited number of cutting edges (ch. 5), limits on the quantity to be produced by a pattern (ch. 5), integer solutions (chs.5, 6 & 7) and a limit on the number of orders to be open simultaneously (ch. 7). While many other constraints will arise in different situations, the ease with which those encountered were handled suggests that a pattern enumeration approach has sufficient flexibility to deal with others;

- (b) "It should be possible to consider costs other than those of trim loss".

Reprocessing costs are present in the case study of ch. 5, while the cost of producing off-cuts was encountered in the case studies of chs. 6 & 7. As can be seen from the development of the pattern enumeration methods, it has proved difficult to include additional costs explicitly, although they can be included implicitly. The extent to which such an implicit method is effective can only be judged by a study of the schedules produced and these appear quite reasonable;

- (c) "It should be possible to cover modifications of the problem which help reduce trim".

In Chapter 5 there are tolerances on the quantities demanded, while in Chapters 6 and 7 off-cuts may be produced. It was found to be quite easy to include both these modifications, which are the two main ones met in practice, within a pattern enumeration method;

- (d) "The method should be economical in computer usage".

It was not possible in any of the case studies to obtain a direct comparison between computer usage by the pattern enumeration method and that of other methods, as the methods used for comparison had all been run on different machines. However the computer requirements of each pattern enumeration method were quite modest, being at most a tenth of the scheduling savings achieved.

Table 19 Computer requirements of the pattern enumeration methods.

Case study of chapter	Store required (words)	Time/schedule (secs.)
5	5K	8
6	4K	2
7	9K	19

- (e) "The method should be easily understood and applied".

The basic idea of pattern enumeration, of simply listing patterns until an acceptable one is found, can be easily understood, and as has been shown in the development of the methods for each of the case studies, the process of building up a method in any particular situation is relatively simple.

The method then, to a large extent, meets these criteria and while its schedules are not optimal they do appear to be better than both manual methods and alternative heuristics. Pattern enumeration thus appears to be applicable to a reasonably wide class of trim loss problems.

The main characteristic of pattern enumeration is that all non-dominated feasible patterns can be listed systematically. Patterns are not skipped or considered twice. The pattern enumeration method of chapter 7 and the final British Steel heuristic only differ significantly in the technique used to generate patterns, yet there is a 21% difference in scrap between the two methods as a result of pattern enumeration not missing potentially valuable patterns. Similarly the pattern enumeration method used in chapter 6 produced better results than Stainton's method which tended to only consider patterns containing one or two different order lengths.

Although the ability to consider feasible patterns exhaustively was an advantage in these three case studies there may be situations where it would be disadvantageous. For example if one knew in advance that only patterns of a certain type would yield efficient schedules, then a technique which only generated that type may have advantages over pattern enumeration which generated many more patterns only to discard them. However, as these case studies show, pattern enumeration methods can be developed which tend to generate certain types of pattern early in the enumeration and this may lessen the problem.

8.3 The Application of Pattern Enumeration Methods

For an approach to scheduling problems to be generally applicable, it must not only be able to yield efficient schedules for a wide class of problems, but it must also be suitable for the range of firms which may wish to use it. Large firms are likely to have analysts and programmers who can develop a pattern enumeration method along the lines indicated. The three case studies show that, given this expertise, the development of an effective method is not difficult.

Small firms, on the other hand, are unlikely to have this expertise. The ideal for them would be a standard package which could be run without any modifications being required. Unfortunately, because of the wide range of characteristics of trim problems, this is at present an unobtainable ideal. Pattern enumeration methods have to be built up step by step around the characteristics of the individual

problem. Nevertheless, it may be possible to develop a package which could be helpful. If, instead of aiming to program a complete method to generate schedules (which would of necessity be limited to only a very narrow class of problems), only the pattern enumeration stage was programmed, then this could be used interactively by manual schedulers tackling a wide range of trim loss problems. The scheduler could set up the list of order sizes in the sequence he wanted and could input acceptance criteria. The computer could then generate patterns until one satisfying the scheduler's criteria was found. It would then be up to the scheduler to decide whether to accept that pattern or let the computer continue the search. Once a pattern acceptable to the scheduler was found then he could decide how much material to cut using that pattern and the computer could update the order list accordingly. Such a package would place all decisions in the hands of the scheduler allowing him to mould the schedules to those required. It would have the advantage over unassisted manual scheduling of enabling a far wider range of patterns to be considered in a systematic manner - and hence more efficient schedules to be found. Also by relieving the scheduler of arithmetic drudgery, it allows the scheduling to be done much more quickly. Such a package could be made available either from a time sharing bureau or (considering its small storage requirements) on a desktop mini-computer. It should not take long for a scheduler to master handling a package of this type.

Whether pattern enumeration is used as part of a complete computer scheduling routine or as a computer aid to manual scheduling, it is important that it uses no more computer time than is absolutely necessary. In §4.5, p.40 several techniques were examined to improve computational efficiency. Of these, one (the use of highest common factors) was particularly effective. When tested in each of the three case studies it produced drops in computing time by at least a factor of three. It is therefore recommended that this technique should be considered in every practical application of pattern enumeration.

8.4 Suggestions for further research

In situations where either optimizing methods (i.e. linear programming and branch & bound) or pattern enumeration could be used it would be valuable to know whether the improvements in scheduling efficiency achieved by optimizing methods would be sufficient to cover the increased computing costs. None of the case studies considered in

this thesis could be formulated for solution by optimizing methods and so it was not possible to carry out this particular assessment of pattern enumeration.

In § 8.3 above, the possibility of using pattern enumeration as an aid to manual scheduling was suggested. In order to test the usefulness of this idea it would be necessary to develop a computer package along the lines described, and for a scheduler to use it in practice. Such an exercise would show how easily such a package could be mastered by a scheduler, whether it really did help him produce more efficient schedules and whether it reduced the time needed to schedule a set of orders.

Throughout this thesis only one-dimensional trim loss problems have been considered. However the steps proposed for developing a scheduling method could equally well be used to yield methods for two-dimensional trim loss problems if a technique can be found to enumerate two-dimensional patterns. Obviously there will be far more two-dimensional patterns than one-dimensional, but provided the pattern acceptance criteria are set sensibly then there will be no need to enumerate them all. The key step in developing such methods will be to define a lexicographic ordering of all patterns so that they can be enumerated systematically.

In considering techniques to improve the computational efficiency of pattern enumeration, the use of a technique based on highest common factors was found to be very effective. Essentially, highest common factors were used to set a bound on the efficiency of a class of patterns so that they could be skipped, i.e. a simple branch and bound procedure was being used. Linear efficiency constraints in integer variables arise in other branch and bound problems (e.g. certain knapsack problems). It is thus possible that the highest common factor techniques, developed within this thesis, may be useful in other situations than the solution of trim loss problems.

8.5 Summary

This thesis has examined the characteristics of one-dimensional trim loss problems met in practice, and different scheduling methods have been considered. While it was not possible to suggest a single method which could be applied to the wide range of these problems, an approach has been developed by which scheduling methods can be built up for a large class of trim loss problems. The approach depends on

being able to tailor the patterns generated at each stage of a scheduling procedure to those which will not lead to intractable residual problems later.

The essential steps of the approach proposed are:

- (a) Schedule a set of orders using a very simple pattern enumeration method (such as that of Fig. 15, p.73);
- (b) Examine the schedules produced to identify those characteristics of the order sizes which tend to lead to intractable residual problems;
- (c) Modify the pattern enumeration method so that the awkward sizes identified in (b) are likely to be scheduled early in the procedure. One way in which this can be achieved is to place awkward order sizes at the head of the list for enumeration. (Other authors, e.g. Johns (13) and Pierce (16), have suggested that there is a 'best' order sequence for all problems. The experience of the case studies of chapters 5, 6 and 7 tends to show that this is not so. The most suitable order sequence will depend on the characteristics of the individual problem);
- (d) Re-schedule the orders using the modified method. Examine the schedules; if unacceptable residual problems remain return to step (b);
- (e) Adjust the pattern acceptance parameters to yield schedules in which a reasonable balance is struck between the various production costs.

While this is a quite simple procedure it yielded scheduling methods which generated better schedules than either manual methods or alternative heuristics. The adaptability of the approach to different problems is a consequence of the ease with which patterns can be moulded to those required within the pattern enumeration routine. The improvement in efficiency of the schedules over those from other heuristics can be attributed to the fact that patterns are considered in a systematic manner. None are missed. Pattern enumeration methods seem to offer a non-optimal, but reasonably efficient, means of scheduling a wide variety of trim loss problems.

Appendix 1. Enumeration Routines

1.1 Data

Data 1 - Parent size = 12000

	ORDER LENGTH	QUANTITY REQUIRED
1	939	8
2	1200	32
3	1000	51
4	1225	31
5	530	20
6	1450	164
7	5900	10
8	300	36
9	5075	38
10	725	4
11	2425	4
12	2100	16
13	3300	28
14	4000	214

Data used to demonstrate the pattern enumeration technique.

1.2 Programs

Program 1:- program segments to enumerate the set of non-dominated patterns until an acceptable one is found.

```

SUBROUTINE PATTERN(M,IL,IS,IWW,ITM,IW)
DIMENSION IL(M),IS(M)
CALL SORT(M,IL)
IW=IWW
JP=0
1  JP=JP+1
   IS(JP)=IW/IL(JP)
   IW=IW-IS(JP)*IL(JP)
   IF(JP.LT.M)GOTO 1
   IF(IW.LE.ITM)RETURN
2  IF(IS(JP).GT.0)GOTO 3
   JP=JP-1
   IF(JP.GT.0)GOTO 2
   GOTO 5
3  IF(JP.EQ.M)GOTO 4
   IS(JP)=IS(JP)-1
   IW=IW+IL(JP)
   GOTO 1
4  IW=IW+IS(M)*IL(M)
   IS(M)=0
   GOTO 2
5  WRITE(2,101)
   STOP
101 FORMAT(' NO ACCEPTABLE PATTERN FOUND')
END
SUBROUTINE SORT(M,IL)
DIMENSION IL(M)
IF(M.LE.1)RETURN
MM=M-1
1  K=0
   DO 2 I=1,MM
     IF(IL(I).GE.IL(I+1))GOTO 2
```



```
K=IL(I)
IL(I)=IL(I+1)
IL(I+1)=K
K=I
2 CONTINUE
MM=K-1
IF(K.GT.1)GOTO 1
RETURN
END
```

Parameters: 1 The number of sizes (m) in the demand list.
2 Array of size m containing the sizes demanded.
3 Array of size m which will contain the pattern chosen.
4 Parent size.
5 Maximum acceptable trim.
6 Trim of the pattern chosen.

Program 2:- Segment to enumerate all non-dominated patterns with a limited number of sizes per pattern.

```
SUBROUTINE PATTERN(M,IL,ID,IS,IWW,ITM,IW)
DIMENSION IL(M),IS(M),ID(M)
CALL SORT(M,IL,ID)
NC=6
IW=IWW
JP=0
1 JP=JP+1
IS(JP)=MIN0(IW/IL(JP),NC,ID(JP))
IW=IW-IS(JP)*IL(JP)
NC=NC-IS(JP)
IF(JP.LT.M)GOTO 1
IF(IW.LE.ITM)RETURN
2 IF(IS(JP).GT.0)GOTO 3
JP=JP-1
IF(JP.GT.0)GOTO 2
GOTO 5
3 IF(JP.EQ.M)GOTO 4
IS(JP)=IS(JP)-1
NC=NC+1
IW=IW+IL(JP)
GOTO 1
4 IW=IW+IS(M)*IL(M)
NC=NC+IS(M)
IS(M)=0
GOTO 2
5 WRITE(2,101)
STOP
101 FORMAT(' NO ACCEPTABLE PATTERN FOUND')
END
```

Parameters are as above for program 1. The maximum number of widths per pattern (NC) has been preset to 6 in this case.

Program 3:- Segment to enumerate Patterns, a check being made that the remainder of the parent size is not smaller than the smallest demand size.

```
SUBROUTINE PATTERN(M,IL,IS,IWW,ITM,IW)
DIMENSION IL(M),IS(M),ID(M)
LSIZE=IWW
DO 6 I=1,M
IS(I)=0
6 IF(IL(I).LT.LSIZE)LSIZE=IL(I)
IW=IWW
JP=0
1 JP=JP+1
IS(JP)=MIN0(IW/IL(JP),ID(JP))
IW=IW-IS(JP)*IL(JP)
IF(JP.LT.M.AND.IW.GE.LSIZE)GOTO 1
IF(IW.LE.ITM)RETURN
2 IF(IS(JP).GT.0)GOTO 3
JP=JP-1
IF(JP.GT.0)GOTO 2
GOTO 5
3 IF(JP.EQ.M)GOTO 4
IS(JP)=IS(JP)-1
IW=IW+IL(JP)
GOTO 1
4 IW=IW+IS(M)*IL(M)
IS(M)=0
GOTO 2
5 WRITE(2,101)
STOP
101 FORMAT(' NO ACCEPTABLE PATTERN FOUND')
END
```

Parameters are as above for program 1.

Program 4:- Segment to enumerate patterns, a check being made that the remainder of the parent size is not smaller than all the remaining sizes on the demand list.

```
SUBROUTINE PATTERN(M,IL,IS,IWW,ITM,IW)
DIMENSION IL(M),IS(M),ID(M),LSIZE(40)
LSIZE(M)=IWW
DO 6 I=2,M
J=M+1-I
IS(I)=0
6 LSIZE(J)=MIN0(LSIZE(J+1),IL(J+1))
IW=IWW
JP=0
1 JP=JP+1
IS(JP)=MIN0(IW/IL(JP),ID(JP))
IW=IW-IS(JP)*IL(JP)
IF(JP.LT.M.AND.IW.GE.LSIZE(JP))GOTO 1
IF(IW.LE.ITM)RETURN
2 IF(IS(JP).GT.0)GOTO 3
JP=JP-1
IF(JP.GT.0)GOTO 2
GOTO 5
3 IF(JP.EQ.M)GOTO 4
IS(JP)=IS(JP)-1
IW=IW+IL(JP)
GOTO 1
```

```
4      IW=IW+IS(M)*IL(M)
      IS(M)=0
      GOTO 2
5      WRITE(2,101)
      STOP
101    FORMAT(' NO ACCEPTABLE PATTERN FOUND')
      END
```

Parameters are as above for program 1.

Program 5:- Segment to enumerate patterns, a check being made on the highest common factor of the sizes remaining on the demand list.

```
      SUBROUTINE PATTERN(M,IL,ID,IS,IWW,ITM,IW)
      DIMENSION IL(M),IS(M),ID(M),ICF(40)
      M1=M+1
      KB=IL(M)
      DO 6 I=1,M
      IS(I)=0
      J=M1-I
      K=KB
      KB=IL(J)
7      KR=KB-K*(KB/K)
      KB=K
      K=KR
      IF(K.NE.0)GOTO 7
6      ICF(J)=KB
      IW=IWW
      JP=0
1      JP=JP+1
      K=ICF(JP)
      IF(IW-K*(IW/K).GT.ITM)GOTO 2
      IS(JP)=MIN0(IW/IL(JP),ID(JP))
      IW=IW-IS(JP)*IL(JP)
      IF(JP.LT.M)GOTO 1
      IF(IW.LE.ITM)RETURN
2      IF(IS(JP).GT.0)GOTO 3
      JP=JP-1
      IF(JP.GT.0)GOTO 2
      GOTO 5
3      IF(JP.EQ.M)GOTO 4
      IS(JP)=IS(JP)-1
      IW=IW+IL(JP)
      GOTO 1
4      IW=IW+IS(M)*IL(M)
      IS(M)=0
      GOTO 2
5      WRITE(2,101)
      STOP
101    FORMAT(' NO ACCEPTABLE PATTERN FOUND')
      END
```

Parameters are as above for program 1.

Program 6:- Segment to enumerate patterns by starting with only one item on the demand list and then increasing the list by one item at a time. The added item to be placed at the top of the list to eliminate repetition of patterns.

```
SUBROUTINE PATTERN(M,IL,ID,IS,IWW,ITM,IW)
DIMENSION IL(M),IS(M),ID(M)
CALL SORT(M,IL,ID)
M=1
IW=IWW
JP=0
1  JP=JP+1
   IS(JP)=IW/IL(JP)
   IW=IW-IS(JP)*IL(JP)
   IF(JP.LT.M)GOTO 1
   IF(IW.LE.ITM)RETURN
2  IF(IS(JP).GT.0)GOTO 3
   JP=JP-1
   IF(JP.GT.0)GOTO 2
   GOTO 5
3  IF(JP.EQ.M)GOTO 4
   IF(JP.EQ.1.AND.IS(1).EQ.1)GOTO 5
   IS(JP)=IS(JP)-1
   IW=IW+IL(JP)
   GOTO 1
4  IW=IW+IS(M)*IL(M)
   IS(M)=0
   GOTO 2
5  WRITE(2,101)
   KD=ID(M+1)
   KL=IL(M+1)
   DO 6 I=1,M
     J=M+1-I
     IL(J+1)=IL(J)
6  ID(J+1)=ID(J)
   ID(1)=KD
   IL(1)=KL
   M=M+1
   JP=0
   IW=IWW
   GOTO 1
101 FORMAT(' NO ACCEPTABLE PATTERN FOUND')
END
```

Parameters are as above for program 1.

Appendix 2. The Roll Slitting Problem

2.1 Data

The data below is drawn from Haessler's case study (11), However some of the results for the multiple pass program are inconsistent with his data. In the data sets which are inconsistent the specific anomaly has been given with the data below and the data has not been used in the current study. There are other inconsistencies with the results of the manual method and the single pass procedure but they have been overlooked as the main comparison is to be made between the pattern enumeration approach and the multiple pass procedure.

Data 2:- Haessler's data set 1 from the initial set of problems.

PARENT SIZE= 197

	ORDER LENGTH	NUMBER
1	11.000	3
2	12.750	6
3	13.250	13
4	13.875	16
5	15.125	40
6	17.250	3
7	17.500	35
8	18.750	10
9	20.000	4
10	20.500	73
11	21.500	13
12	23.750	28
13	25.750	40
14	30.000	256

Data 3:- Haessler's data set 2 from the initial set of problems.

PARENT SIZE= 197

	ORDER LENGTH	NUMBER
1	12.250	62
2	12.625	43
3	15.125	36
4	19.250	8
5	20.000	5
6	21.500	13
7	25.750	6
8	27.500	22
9	30.000	101
10	30.250	36

Data 4:- Haessler's data set 3 from the initial set of problems.

PARENT SIZE= 197

	ORDER LENGTH	NUMBER
1	11.000	4
2	12.250	15
3	13.250	5
4	13.750	11
5	14.000	21
6	14.875	5
7	15.000	36
8	15.750	5
9	16.375	22
10	17.250	2
11	18.000	2
12	25.250	15

Data rejected as -

Total amount cut = Nos. rolls x Roll width = 12 x 197 = 2364
but Total amount output = Demand + Net overproduction + Trim
= 2241.875 + 82 + 38 = 2361.875

Also the total amount of tolerance on the demand is 45.375
while the overproduction is stated as 82.

Data 5:- Haessler's data set 4 from the initial set of problems.

PARENT SIZE= 197

	ORDER LENGTH	NUMBER
1	12.250	11
2	14.500	22
3	14.875	11
4	15.125	21
5	17.500	66
6	20.375	34
7	23.500	14
8	26.000	49
9	30.000	169

Data 6:- Haessler's data set 5 from the initial set of problems.

PARENT SIZE= 197

	ORDER LENGTH	NUMBER
1	11.125	220
2	12.750	16
3	15.125	1
4	30.000	80

Data 7:- Haessler's data set 6 from the initial set of problems.

PARENT SIZE= 197

	ORDER LENGTH	NUMBER
1	13.250	4
2	17.500	9
3	27.250	28
4	29.750	38

Data 8:- Haessler's data set 7 from the initial set of problems.

PARENT SIZE= 197

	ORDER LENGTH	NUMBER
1	12.000	10
2	15.125	21
3	17.000	16
4	17.500	15
5	26.000	11
6	30.000	53

Data rejected as the stated overproduction of 73 can not be made up from the widths on which there exists a tolerance.

Data 9:- Haessler's data set 8 from the initial set of problems.

PARENT SIZE= 197

	ORDER LENGTH	NUMBER
1	11.750	14
2	12.000	8
3	12.844	3
4	13.625	7
5	14.875	79
6	17.000	88
7	18.875	9
8	21.250	13
9	21.375	8
10	27.375	19
11	29.875	117
12	32.000	6

Data rejected as

Total amount cut = Nos. rolls x Roll width = 41 x 197 = 8077

but Total amount output = Demand + Net overproduction + Trim
= 7890.157 + 106 + 78 = 8074.157.

Data 10:- Haessler's data set 9 from the initial set of problems.

PARENT SIZE= 197

	ORDER LENGTH	NUMBER
1	12.250	9
2	13.250	5
3	13.750	39
4	14.875	22
5	19.875	13
6	26.125	10
7	29.875	56
8	30.125	6
9	44.000	4

Data rejected as the total amount of tolerance on the demand is 88.375 while the overproduction is stated to be 114.

Data 11:- Haessler's data set 10 from the initial set of problems.

PARENT SIZE= 197

	ORDER LENGTH	NUMBER
1	12.000	12
2	12.250	62
3	12.625	9
4	12.750	5
5	13.250	10
6	13.875	12
7	15.125	82
8	15.750	82
9	16.250	12
10	17.000	4
11	17.250	11
12	17.500	11
13	17.750	59
14	18.125	4
15	18.750	4
16	18.875	15
17	19.250	1
18	19.500	1
19	19.875	9
20	20.000	11
21	21.500	123
22	23.500	2
23	23.750	8
24	25.750	54
25	26.000	13
26	27.250	14
27	27.500	34
28	29.875	124
29	30.000	327
30	34.250	39
31	44.000	23

Data rejected as

Total amount cut = Nos. rolls x Roll width = 139 x 197 = 27383

but Total amount output = Demand + Net overproduction + Trim
= 28261.125 + 144 + 190 = 28595.125.

Data 12:- Haessler's data set 11 from the initial set of problems.

PARENT SIZE= 197

	ORDER LENGTH	NUMBER
1	11.000	18
2	11.125	7
3	12.000	3
4	12.250	48
5	12.500	1
6	12.625	3
7	12.750	4
8	13.250	3
9	13.875	6
10	14.000	17
11	14.125	6
12	15.125	28
13	15.250	19
14	15.750	5
15	16.125	8
16	16.500	27
17	17.125	23
18	17.250	28
19	17.750	4
20	18.750	4
21	19.125	11
22	19.500	6
23	20.000	2
24	21.375	9
25	23.625	17
26	23.750	16
27	24.375	16
28	25.125	8
29	25.750	5
30	26.125	21
31	27.375	22
32	27.500	21
33	29.750	50
34	29.875	133
35	30.000	198
36	44.000	4

Data rejected as

Total amount cut = Nos. rolls x Roll width = 99 x 197 = 19503

but Total amount output = Demand + Net overproduction + Trim
= 19202.75 + 149 + 99 = 19450.75.

Data 13:- Haessler's data set 12 from the initial set of problems.

PARENT SIZE= 197

	ORDER LENGTH	NUMBER
1	13.250	10
2	13.875	28
3	14.875	135
4	15.125	53
5	15.570	1
6	16.250	10
7	17.000	9

8	17.250	8
9	17.750	4
10	18.125	12
11	18.750	8
12	20.000	1
13	21.500	50
14	23.750	28
15	25.750	7
16	26.000	4
17	27.250	6
18	27.500	24
19	29.750	40
20	30.000	186

Data 14:- Haessler's data set 13 from the initial set of problems.

PARENT SIZE= 197

	ORDER LENGTH	NUMBER
1	11.000	3
2	11.750	3
3	12.750	8
4	13.625	7
5	14.875	44
6	15.000	11
7	16.000	8
8	17.125	4
9	17.782	11
10	19.250	5
11	24.250	13
12	29.750	182
13	30.000	106
14	30.250	2
15	43.000	3

Data 15:- Haessler's data set 14 from the initial set of problems.

PARENT SIZE= 197

	ORDER LENGTH	NUMBER
1	11.000	6
2	12.000	6
3	15.125	82
4	16.125	18
5	16.250	4
6	17.125	23
7	17.250	4
8	17.500	10
9	18.750	4
10	19.875	10
11	20.000	12
12	21.500	7
13	23.500	6
14	23.750	21
15	24.375	16

16	25.125	16
17	25.750	10
18	26.000	9
19	27.375	8
20	29.750	53
21	30.000	449
22	32.000	9

Data 16:- Haessler's data set 15 from the initial set of problems.

PARENT SIZE= 197

	ORDER LENGTH	NUMBER
1	10.000	10
2	11.000	28
3	11.125	6
4	12.125	6
5	12.250	19
6	12.625	9
7	13.250	14
8	13.625	11
9	13.750	8
10	13.875	26
11	14.125	8
12	14.625	20
13	14.875	34
14	15.000	45
15	15.125	70
16	16.125	7
17	16.250	10
18	16.875	7
19	17.125	21
20	17.250	13
21	17.500	6
22	18.875	6
23	19.250	5
24	19.875	18
25	21.375	26
26	23.500	5
27	23.625	12
28	24.375	15
29	25.125	16
30	25.938	7
31	26.125	8
32	27.375	7
33	29.875	128
34	30.000	55
35	30.250	36
36	32.000	9

Data rejected as

Total amount cut = Nos. rolls x Roll width = 78 x 197 = 15366

but Total amount output = Demand + Net overproduction + Trim
= 15152.191 + 107 + 109 = 15368.191.

Data 17:- Haessler's data set 16 from the initial set of problems.

PARENT SIZE= 197

	ORDER LENGTH	NUMBER
1	8.750	4
2	11.000	4
3	11.750	6
4	12.000	6
5	15.125	21
6	15.750	10
7	17.375	27
8	18.750	4
9	25.500	1
10	27.625	20
11	29.750	7
12	30.000	277
13	30.250	31

Data 18:- Haessler's data set 17 from the initial set of problems.

PARENT SIZE= 197

	ORDER LENGTH	NUMBER
1	12.000	8
2	12.750	25
3	14.875	32
4	16.250	85
5	17.500	66
6	21.500	14
7	25.937	5
8	27.500	13
9	29.750	76
10	29.937	11
11	30.000	118
12	30.250	5

Data rejected as

Total amount cut = Nos. rolls x Roll width = 54 x 197 = 10638

but Total amount output = Demand + Net overproduction + Trim
= 10496.742 + 103 + 34 = 10633.742.

Data 19:- Haessler's data set 18 from the initial set of problems.

PARENT SIZE= 197

	ORDER LENGTH	NUMBER
1	12.000	20
2	12.250	39
3	12.625	7
4	12.750	6
5	13.250	20
6	13.875	10
7	14.625	22
8	15.125	34
9	15.750	13

10	16.250	14
11	17.125	8
12	17.250	24
13	17.500	23
14	17.750	4
15	18.125	6
16	18.750	12
17	19.625	10
18	19.750	4
19	21.500	53
20	23.500	10
21	23.615	4
22	23.750	61
23	24.500	39
24	25.125	8
25	25.625	12
26	25.750	149
27	26.000	9
28	27.500	13
29	29.750	37
30	29.875	27
31	30.000	375
32	34.250	37

Data 20:- Haessler's data set 19 from the initial set of problems.

PARENT SIZE= 197

	ORDER LENGTH	NUMBER
1	12.000	15
2	12.250	10
3	12.750	5
4	13.250	10
5	16.250	4
6	17.500	10
7	18.750	10
8	19.250	11
9	19.750	5
10	21.500	17
11	23.500	3
12	23.750	50
13	25.750	3
14	25.937	6
15	27.500	51
16	29.750	56
17	30.000	144

Data 21:- Haessler's data set 20 from the initial set of problems.

PARENT SIZE= 197

	ORDER LENGTH	NUMBER
1	15.125	63
2	19.250	8
3	20.000	5
4	25.750	6
5	27.500	11
6	30.000	31

Data rejected as

Total amount cut = Nos. rolls x Roll width = 14 x 197 = 2758

but Total amount output = Demand + Net overproduction + Trim
= 2593.875 + 109 + 53 = 2755.875.

Also the total amount of tolerance on the demand is 60.25
while the overproduction is stated to be 109.

2.2 Programs

Program 7:- Final program for the solution of the roll slitting
problem.

```
MASTER
COMMON /BLA/M,IL(40),ID(40),IDM(40),ITOL(40),IP(40),
IIT,NWS,TT,NRC,SD,SDL
C INPUT
  READ(1,101)M
  SD=0.0
  SDL=0.0
  DO 14 I=1,M
  READ(1,102)W,ID(I)
  SD=SD+ID(I)
  SDL=SDL+W*ID(I)*1E3
  J=ID(I)/20
  ITOL(I)=4.98-4.0*EXP(-0.15*J)
14  IL(I)=W*1000+1E-1
  READ(1,102)W
  IWW=W*1000+1E-1
  WRITE(2,103)IWW,(I,IL(I),ID(I),ITOL(I),I=1,M)
  IT=1000000
  CALL SORT
C PATTERNS CUT ENTIRELY ON THE MACHINE WINDER
  CALL SEARCHPATYPE(6,IWW,0,2000,9,5)
  CALL SEARCHPATYPE(5,IWW,0,2000,9,0)
C PATTERNS REQUIRING 1 ROLL TO BE SLIT ON THE REPROCESSING WINDER
  CALL SEARCHPATYPE(5,IWW,1500,2000,11,4)
  CALL SEARCHPATYPE(4,IWW,1500,2000,11,0)
C PATTERNS CUT ENTIRELY ON THE REPROCESSING WINDER
  CALL SEARCHPATYPE(2,IWW,3000,2000,14,0)
  CALL SEARCHPATYPE(2,IWW,4500,2000,21,0)
C PATTERNS USING BITS LEFT OVER
  CALL LEFTOVERS(IWW)
  WRITE(2,103)IWW,(I,IL(I),ID(I),ITOL(I),I=1,M)
  TC=TT*1.89+NRC*15
  WRITE(2,104)TT,NRC,TC
  STOP
101  FORMAT(I0)
102  FORMAT(F0.0,I0)
103  FORMAT(' PARENT SIZE=',I7//14X,'LENGTH NUMBER'/(I3,I17,2I6))
104  FORMAT(' TOTAL TRIM=',F14.3/' NUMBER REPROCESSED=',I2
1/' TOTAL COST=',F14.3)
END
SUBROUTINE SEARCHPATYPE(IDUMIN,IWW,NECTR,ITMAX,NWMAX,NLMIN)
COMMON /BLA/M,IL(40),ID(40),IDM(40),ITOL(40),IP(40),
IIT,NWS,TT,NRC,SD,SDL
WRITE(2,101)IDUMIN,NWMAX,NLMIN
ITB=1000000
IDU=ID(1)+1
ISD=IWW
```

```
2   IDU=IDU-1
    IF(IDU.LT.IDUMIN)GOTO 1
    IS=IWW-NECTR
    DO 4 I=1,M
    IF(ID(I).LT.IDU)GOTO 15
    IDM(I)=ID(I)/IDU
    IF(IDM(I)*IDU.NE.ID(I))IDM(I)=(ID(I)-NLMIN)/IDU
4   IS=IS-IL(I)*IDM(I)
    I=M+1
15  IF(IS.GT.ITMAX)GOTO 2
    IF(IS.EQ.ISD)GOTO 2
    ISD=IS
    NWMIN=MIN0(NWMAX,INT(0.99+(SD-NWMAX*(SDL/IWW-IDU))/IDU))
    IF(NLMIN.EQ.0)NWMIN=0
    CALL PATTERN(I-1,IWW-NECTR,NWMIN,NWMAX)
    IF(IT.GE.ITB)GOTO 3
    ITB=IT
    IDS=IDU
    MS=I-1
3   IF(IT.GT.ITMAX)GOTO 2
    IT=IT+NECTR
    SD=SD-NWS*IDU
    SDL=SDL-IDU*FLOAT(IWW-IT)
    CALL OUTPUT(I-1,IDU)
    ITB=1000000
    IDU=MIN0(IDU,ID(I))+1
    ISD=IWW
    GOTO 2
1   IF(IT.GT.ITMAX*3)RETURN
    IT=IT+NECTR
    CALL OUTPUT(MS,IDS)
    RETURN
101  FORMAT(' SEARCHING FOR PATTERNS WITH',I4,' ROLLS','I4,' WI
        IVING',I4)
        END
        SUBROUTINE PATTERN(M,IWW,NWM,NW)
        COMMON /BLA/MF,IL(40),IDF(40),ID(40),ITOL(40),IP(40),
        IIT,NWS,TT,NRC,SD,SDL
        NL=NW-NWM
        COMMON /BLB/IS(40),LSIZE(40),ICF(40)
        N=NW
        M1=M+1
        LSIZE(M)=IWW
        DO 9 I=2,M
        J=M1-I
9   LSIZE(J)=MIN0(LSIZE(J+1),IL(J+1))
        KB=IL(M)
        DO 6 I=1,M
        IS(I)=0
        J=M1-I
        K=KB
        KB=IL(J)
7   KR=KB-K*(KB/K)
        KB=K
        K=KR
        IF(K.NE.0)GOTO 7
6   ICF(J)=KB
        IW=IWW
        JP=0
```



```
1 JP=JP+1
  K=ICF(JP)
  IF(IW-K*(IW/K).GE.IT)GOTO 2
  IS(JP)=MIN0(IW/IL(JP),ID(JP),N)
  IW=IW-IS(JP)*IL(JP)
  N=N-IS(JP)
  IF(JP.LT.M.AND.N.GT.0.AND.IW.GE.LSIZE(JP))GOTO 1
  IF(IW.GE.IT)GOTO 2
  IF(N.GT.NL)GOTO 2
  DO 8 I=1,M
8 IP(I)=IS(I)
  NWS=NW-N
  IT=IW
  IF(IT.LE.0)RETURN
2 IF(IS(JP).GT.0)GOTO 3
  JP=JP-1
  IF(JP.GT.0)GOTO 2
  RETURN
3 IF(JP.EQ.M)GOTO 4
  IS(JP)=IS(JP)-1
  IW=IW+IL(JP)
  N=N+1
  GOTO 1
4 IW=IW+IS(M)*IL(M)
  N=N+IS(M)
  IS(M)=0
  GOTO 2
  END
  SUBROUTINE OUTPUT(MM,IDU)
  COMMON /BLA/M,IL(40),ID(40),IDM(40),ITOL(40),IP(40),
1 IT,NWS,TT,NRC,SD,SDL
  T=IT*1E-3
  WRITE(2,101)IDU,T,NWS
  DO 1 I=1,MM
  IF(IP(I).EQ.0)GOTO 1
  W=IL(I)*1E-3
  WRITE(2,102)IP(I),W
  ID(I)=ID(I)-IP(I)*IDU
1 CONTINUE
  TT=TT+T*IDU
  IF(IDU.LE.4.OR.NWS.GT.9)NRC=NRC+IDU
  IF(IDU.LE.3.OR.NWS.GT.11)NRC=NRC+IDU
  IF(NWS.GT.14)NRC=NRC+IDU*((NWS-8)/7)
  CALL SORT
  IT=1000000
  RETURN
101 FORMAT(/I4,'ROLLS TO BE CUT WITH A TRIM OF',F7.3,' AND',
1 IN THE PATTERN')
102 FORMAT(I4,' OF WIDTH',F7.3)
  END
  SUBROUTINE SORT
  COMMON /BLA/M,IL(40),ID(40),IDM(40),ITOL(40),IP(40),
1 IT,NWS,TT,NRC,SD,SDL
  MM=M-1
1 K=0
  DO 2 I=1,MM
  IF(ID(I).GE.ID(I+1))GOTO 2
  K=ID(I)
  ID(I)=ID(I+1)
  ID(I+1)=K
  K=ITOL(I)
```

```
ITOL(I)=ITOL(I+1)
ITOL(I+1)=K
K=IL(I)
IL(I)=IL(I+1)
IL(I+1)=K
K=I
2 CONTINUE
MM=K-1
IF(K.GT.1)GOTO 1
RETURN
END
SUBROUTINE LEFTOVERS(IWW)
COMMON /BLA/M,IL(40),ID(40),IDM(40),ITOL(40),IP(40),
1 IT,NWS,TT,NRC,SD,SDL
WRITE(2,101)
IS=IWW-3000
N=0
DO 1 I=1,M
ID(I)=ID(I)-ITOL(I)
IS=IS-MAX0(0,ID(I))*IL(I)
1 N=N+MAX0(0,ID(I))
CALL SORT
IF(IS.GE.0)GOTO 3
6 DO 4 I=1,M
IF(ID(I).LE.0)GOTO 5
4 IDM(I)=ID(I)
I=M+1
5 CALL PATTERN(I-1,IWW-3000,0,14)
IT=IT+3000
IS=IS+IWW-IT
CALL OUTPUT(I-1,1)
N=N-NWS
IF(IS.LT.0)GOTO 6
3 DO 7 I=1,M
7 IDM(I)=ITOL(I)*2+MIN0(0,ID(I))
CALL PATTERN(M,IS,0,14)
IT=IT+3000
NWS=NWS+N
DO 2 I=1,M
IP(I)=IP(I)+MAX0(ID(I),0)
2 ID(I)=ID(I)+ITOL(I)
CALL OUTPUT(M,1)
SD=0.0
SDL=0.0
DO 8 I=1,M
IF(ID(I).LT.0)SD=SD-ID(I)*IL(I)*1E-3
8 IF(ID(I).GT.0)SDL=SDL+ID(I)*IL(I)*1E-3
WRITE(2,102)SD,SDL
RETURN
101 FORMAT(' BIT PATTERNS')
102 FORMAT(' OVERPRODUCTION=',F10.3/' UNDERPRODUCTION=',F9.3)
END
FINISH
```

- Input data:-
1. The number of sizes demanded (integer).
 2. The demand list, each width (real) being followed by the quantity required in rolls (integer).
 3. The parent size (real).

2.3 Results

Table 20 Summary of the solutions obtained by manual methods

Data	Number of rolls used	Number of patterns	Net over-production (in)	Number of rolls reprocessed	Trim (in)	Total Cost (\$)
2	68	5	30	8	175	451
3	38	4	116	2	111	240
5	50	5	271	8	123	353
6	27	2	70	29	182	779
7	11	2	0	4	63	179
13	73	7	310	8	193	485
14	56	7	190	10	168	468
15	107	6	341	6	225	515
17	58	6	62	4	87	224
19	141	10	100	20	253	778
20	54	7	27	8	128	362
Total	683	61	1517	107	1708	4834

Table 21 Summary of the solutions obtained by Haessler's single pass method

Data	Number of rolls used	Number of patterns	Net over-production (in)	Number of rolls reprocessed	Trim (in)	Total Cost (\$)
2	68	5	126	8	79	269
3	38	4	116	2	111	240
5	49	5	129	13	68	324
6	27	3	93	28	159	721
7	11	3	16	8	47	209
13	72	7	125	8	181	461
14	56	6	224	13	134	448
15	105	7	-30	12	202	562
17	59	6	208	4	138	321
19	140	11	30	17	126	493
20	54	6	89	8	66	245
Total	679	63	1126	121	1311	4293

Table 22 Summary of the solutions obtained by Haessler's modified single pass method

Data	Number of rolls used	Number of patterns	Net over-production (in)	Number of rolls reprocessed	Trim (in)	Total Cost (\$)
2	68	5	126	8	79	269
3	38	5	173	9	54	237
5	49	5	129	13	68	324
6	27	3	93	28	159	721
7	11	3	16	8	47	209
13	72	7	125	8	181	462
14	56	6	224	13	134	448
15	105	7	-30	12	202	562
17	59	6	208	4	138	321
19	141	10	238	12	115	397
20	54	6	89	8	66	245
Total	680	63	1391	123	1243	4195

Table 23 Summary of the solutions obtained by Haessler's multiple pass method

Data	Number of rolls used	Number of patterns	Net over-production (in)	Number of rolls reprocessed	Trim (in)	Total Cost (\$)
2	68	5	126	8	79	269
3	38	5	167	6	60	203
5	49	6	123	8	74	260
6	27	3	93	28	159	721
7	11	2	0	4	63	179
13	72	7	178	10	128	392
14	55	8	44	8	117	341
15	106	7	174	12	195	549
17	59	6	210	4	138	321
19	140	11	53	12	103	375
20	54	6	89	8	66	245
Total	679	66	1257	108	1182	3855

Table 24 Summary of the solutions obtained by the first pattern enumeration method

Data	Number of rolls used	Number of patterns	Total over-production (in)	Total under-production (in)	Number of rolls reprocessed	Trim (in)	Total Cost (\$)
2	67	8	113.75	140.25	10	34.00	214
3	37	6	60.25	49.75	9	19.625	172
5	48	8	53.00	90.00	6	37.25	160
6	26	3	-	33.375	30	88.75	618
7	11	4	29.75	-	10	33.25	213
13	70	8	23.75	138	6	28.18	143
14	55	8	44.875	-	12	117.398	402
15	104	15	40.875	179.5	22	114.000	545
17	60	11	45.375	77.375	22	577.250	1421
19	140	11	106.00	-	12	51.29	277
20	53	6	-	87.25	12	46.628	268
Total	671	88	517.625	795.5	151	1147.621	4433

(Acceptable trim parameters set to .5", 1.0", .5", 1.0", 2.0", 2.0" for phases (i) - (vi))

Table 25 Summary of the solutions obtained by the pattern enumeration method with a minimum number of pieces in a pattern

Data	Number of rolls used	Number of patterns	Total over-production (in)	Total under-production (in)	Number of rolls reprocessed	Trim (in)	Total Cost (\$)
2	67	8	113.75	140.25	10	34.00	214
3	37	6	37.125	30.00	8	23.00	163
5	48	8	53.00	90.00	6	37.25	160
6	26	3	-	33.375	30	88.75	618
7	11	4	29.75	-	10	33.25	213
13	70	8	23.75	138	6	28.18	143
14	55	8	44.875	-	12	117.398	402
15	104	15	40.875	179.5	22	114.00	545
17	60	11	45.375	77.375	22	577.25	1421
19	139	11	48.75	122.125	10	33.665	214
20	53	7	-	85.00	8	44.378	204
Total	670	89	437.25	895.625	144	1131.121	4297

(Acceptable trim parameters set to .5", 1.0", .5", 1.0", 2.0", 2.0" for phases (i) - (vi)).

Table 26 Summary of the solutions obtained by the pattern enumeration method with modified acceptable trim parameters

Data	Number of rolls used	Number of patterns	Total over-production (in)	Total under-production (in)	Number of rolls reprocessed	Trim (in)	Total Cost (₹)
2	68	8	169.75	17.5	8	52.25	219
3	37	6	-	-	4	30.125	117
5	48	6	-	34.875	6	35.125	156
6	26	3	-	33.375	30	88.75	618
7	11	3	27.25	-	6	35.75	158
13	70	8	23.75	137.75	6	27.93	143
14	55	10	44.875	-	10	117.398	372
15	104	12	-	89.75	10	65.125	273
17	58	15	17.375	147.625	24	281.5	892
19	139	11	-	111.5	8	71.79	256
20	53	7	-	85.00	8	44.378	204
Total	669	89	283.00	657.375	120	850.121	3408

(Acceptable trim parameters set to 1.0", 2.0", 1.0", 2.0", 2.0", 2.0" for phases (i) - (vi))

Table 27 Summary of the solutions obtained by the pattern enumeration technique with the second modified set of acceptable trim parameters

Data	Number of rolls used	Number of patterns	Total over-production (in)	Total under-production (in)	Number of rolls reprocessed	Trim (in)	Total Cost (₹)
2	67	7	47.75	113.75	10	73.50	289
3	37	6	30.00	27.375	8	27.5	172
5	49	6	133.5	-	10	63.75	270
6	26	3	-	33.375	30	88.75	618
7	11	3	27.25	-	6	35.75	158
13	71	8	56.25	29.75	8	84.43	280
14	55	10	44.875	-	10	117.398	372
15	105	7	54.00	47.125	8	165.50	433
17	58	6	73.00	47.375	8	125.625	357
19	140	11	24.25	-	12	133.040	431
20	53	7	-	85.00	8	44.378	204
Total	672	74	490.875	383.750	118	959.621	3584

(Acceptable trim parameters set to 2.0", 2.0", 2.0", 2.0", 2.0", 2.0" for phases (i) - (vi))

Table 28 Comparison of the total costs of different scheduling methods (%)

Method Data	A	B	C	D	E	F	G	H	I
2	451	269	269	269	214	214	219	289	214
3	240	240	237	203	172	163	117	172	117
5	353	324	324	260	160	160	156	270	156
6	779	721	721	721	618	618	618	618	618
7	179	209	209	179	213	213	158	158	158
13	485	461	462	392	143	143	143	280	143
14	468	448	448	341	402	402	372	372	372
15	515	562	562	549	545	545	273	433	273
17	224	321	321	321	1421	1421	892	357	357
19	778	493	397	375	277	214	256	431	214
20	362	245	245	245	268	204	204	204	204
Total	4834	4293	4195	3855	4433	4297	3408	3584	2826

Key to the methods:-

- A - Manual scheduling.
- B - Haessler's single pass method.
- C - Haessler's modified single pass method.
- D - Haessler's multiple pass method.
- E - Simple pattern enumeration method.
- F - Pattern enumeration method with a minimum number of pieces in a pattern.
- G - Pattern enumeration method with a minimum number of pieces in a pattern and modified acceptable trim parameters.
- H - Pattern enumeration method with a minimum number of pieces in a pattern and the second set of modified acceptable trim parameters.
- I - Best result of F, G and H, i.e. pseudo-multiple pass.

Appendix 3. The Bar Cutting Problem

3.1 Data

The seventeen sets of data below are drawn from the company's records. Each represents one week's order, for a particular material and diameter, from one customer.

DATA 22

PARENT SIZE= 480

	ORDER LENGTH	NUMBER
1	248	8
2	205	60
3	202	10
4	201	20
5	200	16
6	196	8
7	173	6
8	138	64
9	135	24
10	45	98

DATA 23

PARENT SIZE= 360

	ORDER LENGTH	NUMBER
1	57	368
2	41	368

DATA 24

PARENT SIZE= 480

	ORDER LENGTH	NUMBER
1	197	64
2	157	64

DATA 25

PARENT SIZE= 480

	ORDER LENGTH	NUMBER
1	215	4
2	207	30
3	206	110
4	200	15
5	199	6
6	188	4
7	150	8
8	142	44
9	141	6
10	139	16
11	122	8
12	118	9
13	114	44
14	110	16
15	108	3
16	79	16
17	51	10
18	49	8
19	47	68

DATA 26

PARENT SIZE= 360

	ORDER LENGTH	NUMBER
1	79	191
2	75	20
3	71	153
4	67	126
5	63	86
6	62	248
7	61	472
8	55	126
9	51	68
10	50	1050
11	47	88
12	43	165
13	41	28
14	27	382
15	26	186

DATA 27

PARENT SIZE= 552

	ORDER LENGTH	NUMBER
1	255	4
2	248	4
3	223	16
4	206	6
5	200	6
6	199	8
7	185	32
8	160	12
9	155	8
10	147	8
11	146	20
12	144	18
13	132	8
14	120	16
15	111	18
16	59	24
17	54	18

DATA 28

PARENT SIZE= 360

	ORDER LENGTH	NUMBER
1	174	13
2	168	7
3	78	41
4	75	22
5	72	88
6	27	42

DATA 29

PARENT SIZE= 360

	ORDER LENGTH	NUMBER
1	282	50
2	270	30
3	252	80
4	246	15
5	240	132
6	228	30

7	207	44
8	204	5
9	201	132
10	186	31
11	177	5
12	174	15
13	165	15
14	156	17
15	144	5
16	141	17
17	129	174
18	120	118
19	108	30
20	105	46
21	102	63
22	96	17
23	94	60
24	90	278
25	81	43
26	78	383
27	72	99
28	69	235
29	66	64
30	57	8
31	48	295
32	45	249
33	42	65
34	39	107
35	36	40

DATA 30

PARENT SIZE= 480

	ORDER LENGTH	NUMBER
1	228	2
2	219	2
3	198	2
4	129	135
5	126	88
6	123	2
7	120	59
8	111	118
9	99	2

DATA 31

PARENT SIZE= 480

	ORDER LENGTH	NUMBER
1	186	11
2	174	6
3	120	12
4	117	18
5	114	20
6	96	6
7	93	4
8	84	22
9	81	8
10	75	8
11	72	24
12	69	16
13	63	32
14	57	8

DATA 32

PARENT SIZE= 552

	ORDER LENGTH	NUMBER
1	234	6
2	210	6
3	138	2
4	120	2

DATA 33

PARENT SIZE= 360

	ORDER LENGTH	NUMBER
1	111	210
2	39	154
3	38	42
4	31	210
5	30	154
6	22	23

DATA 34

PARENT SIZE= 360

	ORDER LENGTH	NUMBER
1	47	42

DATA 35

PARENT SIZE= 360

	ORDER LENGTH	NUMBER
1	256	216
2	197	56
3	89	189
4	79	22
5	39	58
6	37	189
7	26	21
8	18	112
9	12	49

DATA 36

PARENT SIZE= 480

	ORDER LENGTH	NUMBER
1	256	137
2	110	9
3	106	14
4	102	304
5	91	238

DATA 37

PARENT SIZE= 480

	ORDER LENGTH	NUMBER
1	256	136
2	138	240
3	84	52

DATA 38

PARENT SIZE= 480

	ORDER LENGTH	NUMBER
1	256	14

3.2 Programs

Program 8:- Program to solve the bar cropping problem using Gilmore and Gomory's method (10) by initially generating a set of dummy stock lengths (as described in §6.4).

```
MASTER
DIMENSION B(21,21),D(21),P(21),WDEM(20),CDEM(20),IA(20)
100),CS(100) ,A(21),WSC1
COMMON /BLR/RBL(140)/BLI/IX(20,21),ISL(40)
DATA B/441*0.0/,NIT/0/
CALL INPUT(M,NW,WDEM,D,WS,CS)
C SET UP ARRAY B
ML=M+1
DO 1 I=1,ML
1 B(I,I)=1
D(ML)=0
DO 2 I=1,M
B(I,ML)=-CS(I)
2 D(ML)=D(ML)+B(I,ML)*D(I)
C ITERATION
C GENERATE BEST PATTERN IN P
23 DO 3 I=1,M
3 CDEM(I)=-B(I,ML)
CALL GGMANY(WDEM,CDEM,WS,CS,IA,ISL,AMAX,M,NW)
DO 4 I=1,M
4 P(I)=IA(I)
P(ML)=CS(ISL)
A(ML)=-AMAX
IF(A(ML).GE.0)GOTO 8
C FIND LARGEST B.P/D
R=0
DO 16 I=1,M
S=0
DO 14 J=1,ML
14 S=S+B(J,I)*P(J)
A(I)=S
IF(D(I).LE.0)GOTO 16
IF(A(I)/D(I).LE.R)GOTO 16
R=A(I)/D(I)
IP=I
16 CONTINUE
C UPDATE B,IX,D
DO 15 J=1,ML
IX(IP,J)=P(J)
15 P(J)=-A(J)/A(IP)
P(IP)=1/A(IP)
DO 19 J=1,ML
IF(J.EQ.IP)GOTO 19
DO 17 I=1,ML
17 B(I,J)=B(I,J)+P(J)*B(I,IP)
D(J)=D(J)+P(J)*D(IP)
19 CONTINUE
DO 20 I=1,ML
20 B(I,IP)=P(IP)*B(I,IP)
D(IP)=P(IP)*D(IP)
NIT=NIT+1
S=-D(ML)
WRITE(2,108)NIT,S
```



```

      GOTO 23
C PUT INTO INTEGERS
8    CALL MKINT(M,WDEM,D,WS(1))
      STOP
108  FORMAT(' ITERATION',I6,' COST',F15.2)
      END
      SUBROUTINE INPUT(M,NW,WDEM,D,WS,CS)
      DIMENSION WDEM(20),D(21),WS(100),CS(100)
      READ(3,105)C,B
      READ(1,101)M
      READ(1,102)(WDEM(I),D(I),I=1,M)
      READ(1,101)IW
      WS(1)=IW
      CS(1)=IW
      NW=1
      IW=IW-72
1    NW=NW+1
      WS(NW)=IW
      CS(NW)=IW+C*(IW**B)
      IW=IW-6
      IF(IW.GE.WDEM(1))GOTO 1
      WRITE(2,103)(I,WS(I),CS(I),I=1,NW)
      WRITE(2,104)(I,WDEM(I),D(I),I=1,M)
      RETURN
101  FORMAT(10)
102  FORMAT(2F0.0)
103  FORMAT('1',10X,'STOCK'/3X,'LENGTH',10X,'COST'/(I3,F7.0
104  FORMAT(///11X,'ORDERS'/3X,'LENGTH',8X,'NUMBER'/(I3,F6
105  FORMAT(2F0.0)
      END
      SUBROUTINE GGMANY(WDEM,CDEM,WS,CS,IA,ISL,CMA,M,N)
      DIMENSION WDEM(M),CDEM(M),WS(N),CS(N),IA(M)
      COMMON /BLR/W(20),C(20),CM(100)/BLI/IX(20,21),II(20),I
      DO 8 I=1,M
      IA(I)=0
      CMA=-CS(N)
      ISL=N
C SORT AND GENERATE FIRST PATTERN
      MN=0
      IF(CDEM(1).LE.0)GOTO 5
      MN=1
      II(1)=1
      C(1)=CDEM(1)/WDEM(1)
5    DO 1 I=2,M
      IF(CDEM(I).LE.CDEM(I-1))GOTO 1
      IF(CDEM(I).LE.0)GOTO 1
      MN=MN+1
      II(MN)=I
      C(I)=CDEM(I)/WDEM(I)
1    CONTINUE
      MM=MN
      IF(MM.GE.2)GOTO 3
      IF(MM.EQ.0)RETURN
      DO 10 I=1,N
      J=WS(I)/WDEM(II(1))
      IF(J*CDEM(II(1))-CS(I).LE.CMA)GOTO 10
      ISL=I
      CMA=J*CDEM(II(1))-CS(I)
      IA(II(1))=J
10   CONTINUE
      RETURN
3    J=0

```

```
DO 2 I=2,MM
IF(C(II(I-1)).GE.C(II(I)))GOTO 2
J=II(I)
II(I)=II(I-1)
II(I-1)=J
J=I
2 CONTINUE
MM=J-1
IF(J.GT.2)GOTO 3
CC=0
WL=WS(1)
DO 4 I=1,MN
W(I)=WDEM(II(I))
C(I)=CDEM(II(I))
IP(I)=WL/W(I)
WL=WL-IP(I)*W(I)
4 CC=CC+IP(I)*C(I)
WW=WS(1)-WL
DO 7 I=1,N
7 CM(I)=0
C CHOOSE BEST PATTERN
MI=MN+1
IT=1
9 DO 20 I=IT,N
IF(WW.GT.WS(I))GOTO 24
IF(CC.LE.CM(I))GOTO 20
CM(I)=CC
IF(CC-CS(I).LE.CMA)GOTO 20
CMA=CC-CS(I)
DO 22 J=1,MN
22 IA(II(J))=IP(J)
ISL=I
20 CONTINUE
24 DO 6 I=1,MN
J=MI-I
IF(IP(J).NE.0)GOTO 25
6 CONTINUE
RETURN
25 IP(J)=IP(J)-1
CC=CC-C(J)
WW=WW-W(J)
IF(J.EQ.MN)GOTO 9
JI=J+1
DO 23 IT=1,N
IF(WW.GT.WS(IT))GOTO 24
IF((WS(IT)-WW)*C(JI).GT.W(JI)*(CM(IT)-CC))GOTO 21
23 CONTINUE
RETURN
21 WL=WS(IT)-WW
DO 26 I=JI,MN
IP(I)=WL/W(I)
WL=WL-IP(I)*W(I)
26 CC=CC+IP(I)*C(I)
WW=WS(IT)-WL
GOTO 9
END
SUBROUTINE MKINT(M,WDEM,X,WW)
DIMENSION WDEM(M),X(M),D(20),IP(20)
COMMON /BL1/IA(20,21)
DO 9 I=1,M
9 D(I)=0
```

```
      DO 1 I=1,M
      R=X(I)-AINT(X(I))
      X(I)=AINT(X(I))
      DO 1 J=1,M
1      D(J)=D(J)+IA(I,J)*R
      DO 10 I=1,M
10     D(I)=AINT(D(I))
      DO 2 I=1,M
      R=100000
      DO 3 J=1,M
      IF(IA(I,J).EQ.0)GOTO 3
      IF(D(J)/IA(I,J).LT.R)R=D(J)/IA(I,J)
3      CONTINUE
      IF(R.LT.1)GOTO 2
      R=AINT(R)
      X(I)=X(I)+R
      DO 4 J=1,M
4      D(J)=D(J)-R*IA(I,J)
2      CONTINUE
      ST=0
      SS=0
      CALL OUTPUT(M,M,M,X,WDEM,WW,ST,SS)
8      CALL DEMPAT(D,WDEM,M,WW,IP)
      R=100000
      DO 5 I=1,M
      IF(IP(I).EQ.0)GOTO 5
      IF(D(I)/IP(I).LT.R)R=D(I)/IP(I)
5      CONTINUE
      R=AINT(R)
      WRITE(2,103)R
      A=0
      T=WW
      DO 7 I=1,M
      D(I)=D(I)-IP(I)*R
      A=A+D(I)
      T=T-IP(I)*WDEM(I)
7      IF(IP(I).NE.0)WRITE(2,104)IP(I),WDEM(I)
      IF(T.GE.60)GOTO 11
      WRITE(2,101)T
      GOTO 12
11     S=AINT(T/6)*6
      T=T-S
      WRITE(2,102)T,S
      SS=SS+S*R
12     ST=ST+T*R
      IF(A.GE.1)GOTO 8
      WRITE(2,105)ST,SS
      RETURN
101    FORMAT(30X,'TRIM OF',F8.0)
102    FORMAT(30X,'TRIM OF',F8.0,' SEND',F9.0,' TJ STOCK')
103    FORMAT('/' CUT',F10.4,' OF THE FOLLOWING PATTERN')
104    FORMAT(14,' OF WIDTH',F6.1)
105    FORMAT(' TOTAL TRIM=',F24.3/' TOTAL TJ STOCK=',F20.3)
      END
      SUBROUTINE OUTPUT(M,N,NP,X,WDEM,WW,ST,SS)
      DIMENSION X(N),WDEM(M)
      COMMON /BLI/IA(20,21)
      DO 1 I=1,NP
      IF(X(I).LE.1E-4)GOTO 1
```



```
WRITE(2,101)X(I)
T=WW
DO 2 J=1,M
T=T-IA(I,J)*WDEM(J)
2 IF(IA(I,J).NE.0)WRITE(2,102)IA(I,J),WDEM(J)
IF(T.GE.60)GOTO 3
WRITE(2,103)T
GOTO 1
3 S=AINT(T/6)*6
T=T-S
WRITE(2,104)T,S
SS=SS+S*X(I)
1 ST=ST+T*X(I)
RETURN
101 FORMAT(/' CUT',F10.4,' OF THE FOLLOWING PATTERN')
102 FORMAT(I4,' OF WIDTH',F6.1)
103 FORMAT(30X,' TRIM OF',F8.0)
104 FORMAT(30X,' TRIM OF',F8.0,' SEND',F9.0,' TO STOCK')
END
SUBROUTINE DEMPAT(DD,W,M,WST,IP)
DIMENSION DD(M),W(M),IP(M)
COMMON /BLR/D(20),WDEM(20)/BL1/IA(7,20),IS(20),II(20)
WW=WST
WM=WW
N=0
DO 4 I=1,M
IF(DD(I).LE.0.5)GOTO 4
N=N+1
II(N)=I
IS(N)=0
4 IP(I)=0
IF(N.EQ.0)RETURN
IF(N.EQ.1)GOTO 10
NN=N-1
8 K=0
DO 6 I=1,NN
IF(W(II(I)).GE.W(II(I+1)))GOTO 6
K=II(I)
II(I)=II(I+1)
II(I+1)=K
K=I
6 CONTINUE
NN=K-1
IF(K.GT.1)GOTO 8
10 DO 9 I=1,N
D(I)=DD(II(I))
9 WDEM(I)=W(II(I))
JP=0
1 JP=JP+1
IS(JP)=MINI(D(JP),WW/WDEM(JP))
WW=WW-IS(JP)*WDEM(JP)
IF(JP.LT.N)GOTO 1
IF(WW.GE.WM)GOTO 2
WM=WW
DO 5 I=1,N
5 IP(II(I))=IS(I)
IF(WM.LE.10)RETURN
2 IF(IS(JP).GT.0)GOTO 3
JP=JP-1
IF(JP.LE.0)RETURN
GOTO 2
```

```

3   IF(JP.EQ.N)GOTO 7
    IS(JP)=IS(JP)-1
    WW=WW+WDEM(JP)
    GOTO 1
7   WW=WW+IS(N)*WDEM(N)
    IS(N)=0
    GOTO 2
    END
    FINISH

```

Input data:- 1. C and α (real) from input 2.

2. The number of sizes demanded (integer) from input 1.

3. The demand list, each length (real) being followed by the number of bars required (real) from input 1.

4. The parent length (integer) from input 1.

Program 9:- Program to schedule the bar cutting problem by the pattern enumeration method with demand sorted at each stage by size $x \sqrt{\text{demand}}$.

```

MASTER
COMMON IST,ISS,ISN,IW,M,ID(50),IL(50),IP(50),IT
READ(1,101)ND
DO 2 K=1,ND
READ(1,101)ND,M
READ(1,101)(IL(I),ID(I),I=1,M),IW
WRITE(2,103)IW,(I,IL(I),ID(I),I=1,M)
IST=0
ISS=0
ISN=0
1   CALL SORT
    CALL PATTERN
    IF(IT.GT.10)GOTO 3
    CALL OUTPUT
    IF(M.GT.0)GOTO 1
    GOTO 2
4   CALL SORT
3   CALL OFFCUTS
    IT=IT+72
    CALL OUTPUT
    IF(M.GT.0)GOTO 4
2   WRITE(2,102)ND,IST,ISS,ISN
    STOP
101  FORMAT(2I0)
102  FORMAT(I4,/' TOTAL TRIM=',I24/' TOTAL TO STOCK=',I20/'
1    ' TOTAL LENGTHS TO STOCK=',I12)
103  FORMAT(' STOCK LENGTH=',I4/6X,' ORDER LENGTH',8X,' NUMBER'
1    /'(I7,I6,I14))
    END
    SUBROUTINE PATTERN
    COMMON IST,ISS,ISN,IWS,M,ID(50),IL(50),IP(50),IT
    COMMON /BLI/IS(50)
    IW=IWS
    IT=IW*2
    IMAX=MAX0(0,9-M/2)
    JP=0
1    JP=JP+1
    IS(JP)=MIN0(ID(JP),IW/IL(JP))

```

```

IW=IW-IS(JP)*IL(JP)
IF(JP.LT.M)GOTO 1
IF(IW.GE.IT)GOTO 2
IT=IW
DO 5 I=1,M
5 IP(I)=IS(I)
IF(IT.LE.IMAX)RETURN
2 IF(IS(JP).GT.0)GOTO 3
JP=JP-1
IF(JP.GT.0)GOTO 2
RETURN
3 IF(JP.EQ.M)GOTO 4
IS(JP)=IS(JP)-1
IW=IW+IL(JP)
GOTO 1
4 IW=IW+IS(M)*IL(M)
IS(M)=0
GOTO 2
END
SUBROUTINE OUTPUT
COMMON IST,ISS,ISN,IW,M,ID(50),IL(50),IP(50),IT
IS=0
IR=100000
DO 1 I=1,M
IF(IP(I).EQ.0)GOTO 1
IF(ID(I)/IP(I).LT.IR)IR=ID(I)/IP(I)
1 CONTINUE
IF(IT.LT.72)GOTO 4
IS=6*(IT/6)
IT=IT-IS
ISS=ISS+IS*IR
ISN=ISN+IR
4 WRITE(2,101)IR,IT,IS
IST=IST+IR*IT
I=0
2 IF(I.GE.M)RETURN
I=I+1
ID(I)=ID(I)-IR*IP(I)
IF(IP(I).NE.0)WRITE(2,103)IP(I),IL(I)
IF(ID(I).NE.0)GOTO 2
M=M-1
IF(I.GT.M)RETURN
DO 3 J=I,M
J1=J+1
ID(J)=ID(J1)
IL(J)=IL(J1)
3 IP(J)=IP(J1)
I=I-1
GOTO 2
101 FORMAT(' CUT',14,' WITH TRIM OF',14,' AND STOCK OF',14)
103 FORMAT(14,' OF WIDTH',15)
END
SUBROUTINE OFFCUTS
COMMON IST,ISS,ISN,IWS,M,ID(50),IL(50),IP(50),IT
COMMON /BLI/IS(50)
IW=IWS-72
ITT=6
JP=0
1 JP=JP+1
IS(JP)=MIN0(ID(JP),IW/IL(JP))
```



```

IW=IW-IS(JP)*IL(JP)
IF(JP.LT.M)GOTO 1
IWT=IW-6*(IW/6)
IF(IWT.GE.ITT)GOTO 2
ITT=IWT
IT=IW
DO 5 I=1,M
5 IP(I)=IS(I)
IF(ITT.LE.0)RETURN
2 IF(IS(JP).GT.0)GOTO 3
JP=JP-1
IF(JP.GT.0)GOTO 2
RETURN
3 IF(JP.EQ.M)GOTO 4
IS(JP)=IS(JP)-1
IW=IW+IL(JP)
GOTO 1
4 IW=IW+IS(M)*IL(M)
IS(M)=0
GOTO 2
END
SUBROUTINE SORT
COMMON IST,ISS,ISN,IWS,M,ID(50),IL(50),IP(50),IT
COMMON /BLI/R(50)
IF(M.LE.1)RETURN
DO 1 I=1,M
1 R(I)=IL(I)*ID(I)**0.5
MM=M-1
2 K=0
DO 3 I=1,MM
IF(R(I).GE.R(I+1))GOTO 3
S=R(I)
R(I)=R(I+1)
R(I+1)=S
K=ID(I)
ID(I)=ID(I+1)
ID(I+1)=K
K=IL(I)
IL(I)=IL(I+1)
IL(I+1)=K
K=I
3 CONTINUE
MM=K-1
IF(K.GT.1)GOTO 2
RETURN
END
FINISH
```

- Input data:-
1. Data description number (integer).
 2. The number of sizes demanded (integer).
 3. The demand list, each length (integer) being followed by the number of bars required (integer).
 4. The parent length (integer).

Program 10:- The completed program for the bar cutting problem
by pattern enumeration.

```
MASTER
COMMON IST,ISS,ISN,IW,M,ID(50),IL(50),IP(50),IT
READ(1,101)ND
DO 2 K=1,ND
READ(1,101)ND,M
READ(1,101)(IL(I),ID(I),I=1,M),IW
WRITE(2,103)IW,(I,IL(I),ID(I),I=1,M)
IST=0
ISS=0
ISN=0
CALL SORT
1 CALL PATTERN
IF(IT.GT.10)GOTO 3
CALL OUTPUT
IF(M.GT.0)GOTO 1
GOTO 2
3 CALL OFFCUTS
IT=IT+72
CALL OUTPUT
IF(M.LE.0)GOTO 2
IF(IT)1,3,3
2 WRITE(2,102)ND,IST,ISS,ISN
STOP
101 FORMAT(2I0)
102 FORMAT(14,/' TOTAL TRIM=',I24/' TOTAL TO STOCK=',I20 /
1' TOTAL LENGTHS TO STOCK=',I12)
103 FORMAT(' STOCK LENGTH=',I4/6X,' ORDER LENGTH',8X,' NUMBER '
1/(I7,I6,I14))
END
SUBROUTINE PATTERN
COMMON IST,ISS,ISN,IWS,M,ID(50),IL(50),IP(50),IT
COMMON /BLI/IS(50)
IW=IWS
IT=IW*2
JP=0
1 JP=JP+1
IS(JP)=MIN0(ID(JP),IW/IL(JP))
IW=IW-IS(JP)*IL(JP)
IF(JP.LT.M)GOTO 1
IF(IW.GE.IT)GOTO 2
IT=IW
DO 5 I=1,M
5 IP(I)=IS(I)
IF(IT.LE.0)RETURN
2 IF(IS(JP).GT.0)GOTO 3
JP=JP-1
IF(JP.GT.0)GOTO 2
RETURN
3 IF(JP.EQ.1.AND.IS(1).EQ.1)RETURN
IF(JP.EQ.M)GOTO 4
IS(JP)=IS(JP)-1
IW=IW+IL(JP)
GOTO 1
4 IW=IW+IS(M)*IL(M)
IS(M)=0
GOTO 2
END
```

```
SUBROUTINE OUTPUT
COMMON IST,ISS,ISN,IW,M,ID(50),IL(50),IP(50),IT
IS=0
IR=100000
DO 1 I=1,M
IF(IP(I).EQ.0)GOTO 1
IF(ID(I)/IP(I).LT.IR)IR=ID(I)/IP(I)
1 CONTINUE
IF(IT.LT.72)GOTO 4
IS=6*(IT/6)
IT=IT-IS
ISS=ISS+IS*IR
ISN=ISN+IR
4 WRITE(2,101)IR,IT,IS
IST=IST+IR*IT
IF(ID(I).LE.IP(I)*IR)IT=-1
I=0
2 IF(I.GE.M)RETURN
I=I+1
ID(I)=ID(I)-IR*IP(I)
IF(IP(I).NE.0)WRITE(2,103)IP(I),IL(I)
IF(ID(I).NE.0)GOTO 2
M=M-1
IF(I.GT.M)RETURN
DO 3 J=I,M
J1=J+1
ID(J)=ID(J1)
IL(J)=IL(J1)
3 IP(J)=IP(J1)
I=I-1
GOTO 2
101 FORMAT(' CUT',I4,' WITH TRIM OF',I4,' AND STOCK OF',I4)
103 FORMAT(I4,' OF WIDTH',I5)
END
SUBROUTINE OFFCUTS
COMMON IST,ISS,ISN,IWS,M,ID(50),IL(50),IP(50),IT
COMMON /BLI/IS(50)
IW=IWS-72
ITT=6
JP=0
1 JP=JP+1
IS(JP)=MIN0(ID(JP),IW/IL(JP))
IW=IW-IS(JP)*IL(JP)
IF(JP.LT.M)GOTO 1
IWT=IW-6*(IW/6)
IF(IWT.GE.ITT)GOTO 2
ITT=IWT
IT=IW
DO 5 I=1,M
5 IP(I)=IS(I)
IF(ITT.LE.0)RETURN
2 IF(IS(JP).GT.0)GOTO 3
JP=JP-1
IF(JP.GT.0)GOTO 2
RETURN
3 IF(JP.EQ.1.AND.IS(1).EQ.1)RETURN
IF(JP.EQ.M)GOTO 4
IS(JP)=IS(JP)-1
IW=IW+IL(JP)
GOTO 1
4 IW=IW+IS(M)*IL(M)
IS(M)=0
GOTO 2
END
```



```
SUBROUTINE SORT
COMMON IST,ISS,ISN,IWS,M,ID(50),IL(50),IP(50),IT
COMMON /BLI/R(50)
IF(M.LE.1)RETURN
DO 1 I=1,M
R(I)=IL(I)*ID(I)**0.5
1 IF(IWS.EQ.IL(I)*(IWS/IL(I)))R(I)=R(I)-100000
MM=M-1
2 K=0
DO 3 I=1,MM
IF(R(I).GE.R(I+1))GOTO 3
S=R(I)
R(I)=R(I+1)
R(I+1)=S
K=ID(I)
ID(I)=ID(I+1)
ID(I+1)=K
K=IL(I)
IL(I)=IL(I+1)
IL(I+1)=K
K=I
3 CONTINUE
MM=K-1
IF(K.GT.1)GOTO 2
RETURN
END
FINISH
```

Input data as above for program 9.

Program 11:- Program used for the bar cutting problem with non-zero initial stock position. This program uses several of the methods described in §4.5 in order to cut down solution time. In particular the methods (b), (c) and (f) of §4.5 are used.

```
MASTER
COMMON /BLA/JS(30,60),IM(30),ISK(30),ITCH,IST,ISS,KT
READ(1,101)JS
READ(1,101)IM,ISK
READ(4,101)NS,ITCH
WRITE(5)JS
DO 1 I=1,NS
IF(I.EQ.20*(I/20))WRITE(5)JS
1 CALL SCHED
WRITE(5)JS
STOP
101 FORMAT(10I0)
END
SUBROUTINE SCHED
COMMON /BLA/JS(30,60),IM(30),ISK(30),ITCH,IST,ISS,KT
COMMON /BLB/M,ID(400),IL(400),ICD(400),LSIZE
COMMON /BLC/RBL(400)
READ(3)KT,M
KT=KT-10
IF(KT.GT.20)KT=KT-10
M=M+1
```

```
READ(3)(IL(M1-I),I=1,M),(ID(M1-I),I=1,M)
LSIZE=IL(M)
ISS=0
IST=0
SC=0
KB=IL(M)
DO 5 I=1,M
J=M1-I
K=KB
KB=IL(J)
7 KR=KB-K*(KB/K)
KB=K
K=KR
IF(K.NE.0)GOTO 7
ICD(J)=KB
5 SC=SC+IL(I)*ID(I)
ICD(M1)=1
C PATTERNS FROM SCRAP
K=(LSIZE-1)/200
3 K=K+1
IF(K.GT.60)GOTO 10
IF(JS(KT,K).EQ.0)GOTO 3
CALL DPW(K*200,JS(KT,K))
IF(M)20,20,3
C PATTERNS FROM PARENT SIZE
10 K=ISK(KT)
IF(M.LE.1)GOTO 11
I=0
1 I=I+1
IF(I.GT.M)GOTO 6
RBL(I)=IL(I)*ID(I)**0.5
IF(K.EQ.IL(I)*(K/IL(I)))RBL(I)=RBL(I)-100000
GOTO 1
6 MM=M-1
4 K=0
DO 2 I=1,MM
IF(RBL(I).GE.RBL(I+1))GOTO 2
K=IL(I)
IL(I)=IL(I+1)
IL(I+1)=K
K=ID(I)
ID(I)=ID(I+1)
ID(I+1)=K
R=RBL(I)
RBL(I)=RBL(I+1)
RBL(I+1)=R
K=I
2 CONTINUE
MM=K-1
IF(K.GT.1)GOTO 4
11 KB=IL(M)
M1=M+1
DO 8 I=1,M
J=M1-I
K=KB
KB=IL(J)
9 KR=KB-K*(KB/K)
KB=K
K=KR
IF(K.NE.0)GOTO 9
8 ICD(J)=KB
```

```
CALL TW
20 WRITE(2,108)KT,SC,IST,ISS
RETURN
108 FORMAT(I12,F20.0,2I12)
END
SUBROUTINE DPW(IWS,INS)
COMMON /BLB/M,ID(400),IL(400),ICD(400),LSIZE
COMMON /BLC/IW,IS(400),IP(400),JP
9 IW=IWS
DO 12 I=1,M
12 IS(I)=0
IMIN=9
IT=IW*2
JP=1
10 K=ICD(JP)
IF(IW-K*(IW/K).GE.IT)GOTO 2
IS(JP)=MIN0(ID(JP),IW/IL(JP))
IW=IW-IS(JP)*IL(JP)
1 JP=JP+1
IF(IW.LT.LSIZE)GOTO 11
IF(JP-M)10,10,11
11 IF(IW.GE.IT)GOTO 2
IF(IW.GT.IMIN)GOTO 4
CALL RED(INS)
IF(M*INS.EQ.0)RETURN
GOTO 1
4 IT=IW
DO 5 I=1,M
5 IP(I)=IS(I)
2 JP=JP-1
IF(JP.EQ.0)GOTO 6
IF(IS(JP))2,2,3
3 IF(JP.EQ.M)GOTO 7
IS(JP)=IS(JP)-1
IW=IW+IL(JP)
JP=JP+1
GOTO 10
7 IW=IW+IS(M)*IL(M)
IS(M)=0
GOTO 2
6 IW=IWS
IF(IT.GT.50)RETURN
DO 8 I=1,M
IS(I)=IP(I)
IF(IP(I).GT.ID(I))GOTO 9
8 IW=IW-IP(I)*IL(I)
IF(IW.NE.IT)GOTO 9
IMIN=IW
CALL RED(INS)
IF(M*INS.EQ.0)RETURN
GOTO 9
END
SUBROUTINE TW
COMMON /BLA/JS(30,60),IM(30),ISK(30),ITCH,IST,ISS,KT
COMMON /BLB/M,ID(400),IL(400),ICD(400),LSIZE
COMMON /BLC/IW,IS(400),IP(400),JP
9 IW=ISK(KT)
DO 13 I=1,M
13 IS(I)=0
IMIN=4
IT=IW*2
JP=1
1 K=ICD(JP)
```



```
IF(IW-K*(IW/K).GE.IT)GOTO 3
IS(JP)=MIN0(ID(JP),IW/IL(JP))
IW=IW-IS(JP)*IL(JP)
5 JP=JP+1
IF(IW.LT.LSIZE)GOTO 7
IF(JP-M)1,1,7
7. IF(IW.GE.IT)GOTO 3
IF(IW.GT.IMIN)GOTO 8
CALL RED(1000000)
IF(M.EQ.0)RETURN
IF(JP.EQ.0)IMIN=4
GOTO 5
8 IT=IW
DO 2 I=1,M
2 IP(I)=IS(I)
3 JP=JP-1
IF(IS(JP))3,3,4
4 IF(JP.LE.1.AND.IS(1).LE.1)GOTO 11
IF(JP.EQ.M)GOTO 6
IS(JP)=IS(JP)-1
IW=IW+IL(JP)
JP=JP+1
GOTO 1
6 IW=IW+IL(M)*IS(M)
IS(M)=0
IF(M.GT.1)GOTO 3
11 IW=ISK(KT)
IF(IT.GT.ITCH.AND.IL(1).LT.ISK(KT)-IM(KT))GOTO 12
DO 10 I=1,M
IS(I)=IP(I)
IF(IP(I).GT.ID(I))GOTO 9
10 IW=IW-IP(I)*IL(I)
IF(IW.NE.IT)GOTO 9
IMIN=IW
CALL RED(1000000)
IF(JP.EQ.0)IMIN=4
IF(M.EQ.0)RETURN
GOTO 9
12 CALL LEFTS(ISK(KT),IM(KT))
IMIN=4
IF(M.GT.0)GOTO 9
RETURN
END
SUBROUTINE LEFTS(IWS,IMIN)
COMMON /BLB/M,ID(400),IL(400),ICD(400),LSIZE
COMMON /BLC/IWW,IS(400),IP(400),JP
MIN=4
9 IW=IWS-IMIN
DO 8 I=1,M
IS(I)=ID(I)
IW=IW-IL(I)*ID(I)
IF(IW.LT.0)GOTO 15
8 CONTINUE
GOTO 11
15 KB=200
MI=M+1
DO 14 I=1,M
J=MI-I
K=KB
KB=ICD(J)
10 KR=KB-K*(KB/K)
```

```
KB=K
K=KR
IF(K.NE.0)GOTO 10
ICD(J)=KB
14 IS(I)=0
IW=IWS-IMIN
IR=500
JP=1
5 K=ICD(JP)
IF(IW-K*(IW/K).GE.IR)GOTO 3
IS(JP)=MIN0(ID(JP),IW/IL(JP))
IW=IW-IS(JP)*IL(JP)
1 JP=JP+1
IF(IW.LT.LSIZE)GOTO 7
IF(JP-M)5,5,7
7 IRN=IW-200*(IW/200)
IF(IRN.GE.IR)GOTO 3
IF(IRN.LE.MIN)GOTO 11
IT=IW
DO 2 I=1,M
2 IP(I)=IS(I)
IR=IRN
3 JP=JP-1
IF(IS(JP))3,3,4
4 IF(JP.EQ.1.AND.IS(1).EQ.1)GOTO 12
IF(JP.EQ.M)GOTO 6
IS(JP)=IS(JP)-1
IW=IW+IL(JP)
JP=JP+1
GOTO 5
6 IS(M)=IS(M)-1
IW=IW+IL(M)
JP=JP+1
GOTO 7
12 IW=IT
DO 13 I=1,M
13 IS(I)=IP(I)
MIN=IR
11 IWW=IW+IMIN
CALL RED(1000000)
IF(JP.EQ.0)RETURN
GOTO 9
END
SUBROUTINE RED(INS)
COMMON /BLA/JS(30,60),IM(30),ISK(30),ITCH,IST,ISS,KT
COMMON /BLB/M,ID(400),IL(400),ICD(400),LSIZE
COMMON /BLC/IW,IS(400),IP(400),JP
IR=INS
I=M
2 K=IS(I)
IF(K)3,4,3
3 IRR=ID(I)/K
IF(IR-IRR)4,4,5
5 IR=IRR
4 I=I-1
IF(I)9,9,2
9 INS=INS-IR
IF(IW.LT.IM(KT))GOTO 1
K=IW/200
KS=K*200
JS(KT,K)=JS(KT,K)+IR
```

5. For each schedule the following information is required in free integer format on the magnetic tape CARD 5 (input 3).
- (a) Material code.
 - (b) Number of items in the demand list.
 - (c) The demand list, each length (in m.m.) being followed by the number of bars required.

Note:- In the seventeen sets of data all measurements are in imperial units however the data for the week's test run is metric. Hence multiples of .2 m. are sent to stock instead of multiples of 6".


```

IW=IW-KS
ISS=ISS+KS*IR
1  IST=IST+IR*IW
   JP=M
   I=0
12  I=I+1
   IF(I.GT.M)RETURN
   ID(I)=ID(I)-IS(I)*IR
   IF(ID(I).GE.IS(I))GOTO 12
   JP=I-1
   DO 10 J=I,M
10  IW=IW+IS(J)*IL(J)
   IF(ID(I).GT.0)GOTO 11
8   M=M-1
   IF(I.GE.M+1)GOTO 6
   DO 13 J=I,M
   J1=J+1
   ID(J)=ID(J1)
   IL(J)=IL(J1)
13  IS(J)=IS(J1)
   I=I-1
11  I=I+1
   IF(I.GT.M)GOTO 6
   ID(I)=ID(I)-IS(I)*IR
   IS(I)=0
   IF(ID(I))8,8,11
6   LSIZE=12000
   KB=IL(M)
   M1=M+1
   DO 7 J=1,M
   I=M1-J
   IF(IL(I).LT.LSIZE)LSIZE=IL(I)
   K=KB
   KB=IL(I)
14  KR=KB-K*(KB/K)
   KB=K
   K=KR
   IF(K.NE.0)GOTO 14
7   ICD(I)=KB
   RETURN
   END
   FINISH
```

- Input data:-
1. The initial stock position in an array (integer) giving the number of offcuts in sizes 0 - 12 m. in steps of 0.2 m. for 30 material codes from input 1.
 2. The minimum length to be sent to stock for each type (integer) from input 1.
 3. The total number of schedules (integer) from input 4.
 4. The acceptable trim in m.m. (integer) from input 4.

3.3 Results

Key to tables -

A - Stainton's method.

B - Pattern enumeration method sorting at each stage by size.

C - " " " " " " " " " " with regard to off-cuts.

D - " " " " " " " " " demand, " " " "

E - " " " " " " " " size & demand, " " "

F - " " " " " " " " size & $\sqrt{\text{demand}}$ " " "

G - " " " " " once by " " " " " "

H - " " " " " " " " " " " " " "

divisors of the parent length being kept at the foot of the demand list.

Table 29 Total trim incurred by different methods of scheduling the bar cutting problem (in.)

Method Data	A	B	C	D	E	F	G	H
22	480	426	150	246	162	162	156	156
23	866	740	146	146	146	146	146	146
24	258	258	252	252	252	252	0	0
25	870	3 780	612	492	570	630	132	132
26	2 407	1 501	1 327	847	631	601	43	43
27	826	268	76	106	88	58	70	70
28	18	516	96	24	96	84	36	6
29	1 575	483	99	483	357	69	309	375
30	51	63	15	27	27	27	15	15
31	294	186	24	66	18	48	6	6
32	276	66	0	0	0	0	0	0
33	1 144	442	178	286	622	232	64	64
34	0	186	0	0	0	0	0	0
35	1 372	784	706	886	832	748	604	604
36	664	1 924	526	526	526	526	262	262
37	416	4 364	272	272	272	272	272	272
38	28	28	28	28	28	28	28	28
Total	11 545	16 015	4 507	4 687	4 627	3 883	2 143	2 179
Time to run (secs.).	-	16	16	28	24	28	21	20

Table 30 Total amount sent to stock by different methods of scheduling the bar cutting problem. (in.)

Method Data	A	B	C	D	E	F	G	H
22	12108	2562	3318	6582	4746	4746	4752	4752
23	870	276	3750	3750	3750	3750	3750	3750
24	3006	3006	3012	3012	3012	3012	8064	8064
25	41676	366	28974	30534	30936	30876	15534	15534
26	174	72	966	366	222	2412	810	810
27	2316	114	858	828	1950	1428	1416	1416
28	3666	288	1428	2220	708	1440	3288	3678
29	1608	180	1644	36540	8946	954	4674	2088
30	3294	3282	3810	3798	3798	3798	3330	3330
31	162	270	912	390	438	408	450	450
32	408	618	1236	1236	1236	1236	1236	1236
33	5628	210	1914	9366	1110	420	1308	588
34	546	0	546	546	546	546	546	546
35	24096	12444	12882	22062	20676	13200	12264	12264
36	36324	11544	32622	32622	32622	32622	22326	22326
37	53040	18372	37824	37824	37824	37824	17664	17664
38	3108	3108	3108	3108	3108	3108	3108	3108
Total	192030	56712	138804	194784	155628	141780	104520	101604
Average length of off-cut	182	150	164	156	165	172	119	119

Table 31 Total number of lengths sent to stock by different methods of scheduling the bar cutting problem.

Method Data	A	B	C	D	E	F	G	H
22	54	30	35	43	39	39	39	39
23	12	1	50	50	50	50	50	50
24	33	33	33	33	33	33	64	64
25	162	1	109	122	116	114	112	112
26	1	1	12	3	1	20	5	5
27	15	1	7	5	13	6	8	8
28	31	1	11	12	4	11	26	29
29	12	1	20	301	77	6	35	17
30	33	31	34	34	34	34	33	33
31	1	1	5	2	2	1	2	2
32	2	4	8	8	8	8	8	8
33	40	1	22	71	8	2	14	5
34	7	0	7	7	7	7	7	7
35	218	122	125	189	184	126	128	128
36	174	52	174	173	173	173	195	195
37	248	84	178	178	178	178	136	136
38	14	14	14	14	14	14	14	14
Total	1057	378	844	1245	941	822	876	852

Table 32 Comparison of the pattern enumeration method for the bar cutting problem on grouped and ungrouped data.

Material and Diameter		Total amount cut (m)	Trim (m)		Amount to stock (m)		
			Grouped	Ungrouped	grouped	ungrouped	
Mild steel	6mm	24 139	10	124	49	9 517	
"	"	8mm	20 718	1	86	3	3 353
"	"	10mm	70 792	6	237	7	17 560
"	"	12mm	38 334	3	99	19	3 562
High Tensile	6mm	71 158	268	511	1 385	13 952	
"	"	8mm	1 654	5	19	1	817
"	"	10mm	73 861	224	650	207	11 465
Total		300 656	517	1,726	1 671	60 226	

In the table above, a comparison can be made between the trim figures for grouped and ungrouped data. A similar comparison of the amount sent to stock would be misleading, as in the ungrouped data the lengths sent to stock after cutting one customer's order could be taken out of stock again to cut another's. The comparison of the final stock position produced did, however, show that the 'grouped data' had, when cut, given rise to far lower stocks.

Appendix 4. The Sequential Bar Cutting Problem.

4.1 Data

The twenty data sets below are those used by British Steel in developing their heuristic method ((7) and (8)). Each set of data shows the orders for rollings of a particular section and gauge.

DATA 39

ORDER

1	7 OF 16.77	7 OF 18.30		
2	26 OF 17.00	10 OF 16.10	6 OF 15.00	12 OF 13.75
	17 OF 12.90			
3	5 OF 18.27	1 OF 18.27	1 OF 10.41	
4	14 OF 12.80	4 OF 18.28		
5	4 OF 9.80	5 OF 11.00	7 OF 12.80	
6	4 OF 15.60	4 OF 15.30		
7	10 OF 12.00	7 OF 14.00	7 OF 15.00	
8	30 OF 18.28			
9	10 OF 15.24			

FIRST SET OF PARENT LENGTHS

70.31	72.74	64.66	67.30	72.85	67.37	70.55	69.97	64.87	66.43
72.74	64.55	72.59	63.60	63.24	76.03	73.05	73.97	75.35	69.38
71.09	73.09	76.72	62.48	66.37	71.94	71.27	74.14	71.37	72.94
64.27	61.14	69.37	71.95	50.28	69.18	62.47	73.25	70.19	69.88
61.58	60.57	68.17	73.83	76.43	72.10	73.74	73.55	61.53	72.42

SECOND SET OF PARENT LENGTHS

73.36	72.31	70.60	72.78	71.24	63.45	72.52	67.05	64.57	51.95
72.51	77.51	70.44	73.05	67.31	60.41	68.64	66.16	51.15	72.43
50.21	75.35	61.22	72.15	76.87	70.84	75.19	75.56	74.59	50.51
73.69	71.55	71.05	68.38	75.81	70.41	61.14	70.15	76.63	70.35
78.48	63.67	50.67	77.00	67.90	64.43	63.46	61.85	70.96	71.10

DATA 40

ORDER

1	7 OF 21.80	22 OF 18.30	37 OF 15.25	11 OF 12.80
	26 OF 12.25			

FIRST SET OF PARENT LENGTHS

69.27	87.22	65.90	89.35	63.04	61.04	88.84	85.68	86.51	91.95
85.08	62.90	87.67	48.92	64.49	66.64	77.40	69.63	91.12	68.01
65.02	89.97	68.64	90.08	63.71	85.56	65.90	85.38	69.10	86.23

SECOND SET OF PARENT LENGTHS

61.27	65.93	63.12	48.36	64.04	62.99	89.57	69.45	64.56	86.31
68.76	63.76	69.75	61.62	69.94	86.06	47.88	67.75	60.60	62.79
63.34	84.88	85.21	66.38	68.37	86.72	87.99	65.45	67.82	65.84

DATA 41

ORDER

1	10 OF 11.50			
2	8 OF 13.71			
3	1 OF 14.02	1 OF 15.42		
4	24 OF 7.22			
5	2 OF 14.02			
6	3 OF 17.72	5 OF 12.05	1 OF 18.05	3 OF 12.50
	2 OF 18.50			
7	2 OF 12.73			
8	1 OF 4.30			

9	2	DF	15.25	3	DF	18.30						
10	3	DF	10.00	3	DF	15.00						
11	7	DF	12.10									
12	10	DF	12.20									
13	9	DF	7.65									
14	3	DF	15.30	2	DF	18.00						
15	4	DF	10.07	2	DF	8.10	2	DF	7.57			
16	6	DF	12.97						1	DF	7.05	
17	7	DF	6.55									
18	7	DF	11.00	7	DF	13.00	6	DF	15.00	6	DF	18.00
19	6	DF	12.00	7	DF	15.00						
20	3	DF	12.00	9	DF	16.00						
21	3	DF	15.30	4	DF	9.80	3	DF	11.00	3	DF	12.20
	3	DF	14.10									
22	1	DF	11.75	1	DF	8.00	1	DF	7.85	1	DF	8.00
	2	DF	7.85									
23	4	DF	14.00	4	DF	13.33						
24	13	DF	15.30									
25	1	DF	14.05	1	DF	13.40	2	DF	11.62	1	DF	7.77
26	1	DF	5.90									
27	1	DF	13.71									
28	1	DF	8.87	1	DF	8.00	1	DF	10.71			
29	1	DF	13.55	1	DF	8.87	2	DF	8.00	3	DF	5.35
30	1	DF	9.00									
31	15	DF	13.50	13	DF	13.25	12	DF	12.05	4	DF	12.05
32	3	DF	12.55	2	DF	11.65	1	DF	7.05	1	DF	8.85
	3	DF	6.75									
33	3	DF	11.76									
34	1	DF	8.40									
35	1	DF	4.05									
36	4	DF	18.30									
37	9	DF	8.41	10	DF	8.42						
38	8	DF	15.24	15	DF	18.28						
39	20	DF	18.28									
40	13	DF	12.19	9	DF	18.28						
41	5	DF	18.28	5	DF	16.76						
42	1	DF	8.05	1	DF	7.77						
43	1	DF	10.70									
44	3	DF	10.43	8	DF	10.21	2	DF	10.10	3	DF	9.95
	1	DF	6.90	6	DF	10.43	5	DF	10.21	3	DF	10.13
45	4	DF	9.80	2	DF	9.62	3	DF	9.44	1	DF	8.68
	1	DF	9.90	1	DF	8.30	1	DF	6.65			

FIRST SET OF PARENT LENGTHS

79.97	87.83	88.22	83.83	75.97	80.33	85.22	74.33	79.96	84.80
82.14	85.58	86.21	77.05	85.38	82.81	83.39	83.10	59.01	81.20
54.09	83.68	75.26	83.43	88.28	77.75	84.99	77.13	84.86	52.00
85.24	86.39	81.14	77.33	85.70	85.28	84.31	75.33	79.22	82.33
76.98	81.86	82.30	77.07	90.68	84.46	77.60	52.45	76.33	76.90
78.42	76.42	81.75	83.67	77.25	85.40	84.19	74.49	80.20	58.77
77.79	87.78	79.57	85.37	86.09	84.19	81.29	87.02	78.55	85.07
86.45	81.02	87.04	85.03	84.82	81.61	83.31	78.35	86.29	88.52

SECOND SET OF PARENT LENGTHS

86.73	85.49	90.34	64.54	83.24	84.51	88.87	84.65	85.04	85.42
83.11	78.87	85.22	83.45	88.71	75.17	83.63	84.17	82.35	85.60
77.39	85.96	86.25	83.78	74.43	71.59	83.64	64.45	59.95	85.65
80.30	82.94	81.92	82.06	86.04	85.67	85.61	78.62	84.20	84.65
85.05	86.43	87.15	78.99	85.56	81.48	84.78	82.38	81.26	77.06
81.03	77.64	85.52	82.34	81.36	87.13	76.51	77.81	87.27	82.36
75.67	86.81	69.78	85.38	87.25	75.03	65.95	78.38	76.74	83.00
78.36	83.11	76.43	76.59	86.63	74.44	77.91	72.52	85.92	85.81

DATA 42

ORDER

1	5 OF 11.00	6 OF 12.50	6 OF 13.50	
2	5 OF 9.00	5 OF 10.00	4 OF 11.00	4 OF 12.00
	3 OF 16.00			
3	5 OF 9.80	4 OF 11.00	4 OF 12.20	3 OF 14.10
	3 OF 15.30			
4	3 OF 14.60			
5	4 OF 11.65	4 OF 17.50	6 OF 8.15	
6	7 OF 12.00	2 OF 6.60	2 OF 8.55	
7	3 OF 12.20			
8	1 OF 11.05			
9	2 OF 9.50	1 OF 4.30	2 OF 8.85	
10	1 OF 15.24			
11	1 OF 10.47			
12	2 OF 12.05	2 OF 14.37	1 OF 9.35	4 OF 10.25

FIRST SET OF PARENT LENGTHS

92.45 75.56 85.31 89.77 78.81 78.89 95.04 76.21 86.86 94.29
 77.99 67.32 72.94 78.78 77.20 96.20 81.34 88.22 83.28 94.68

SECOND SET OF PARENT LENGTHS

82.86 77.58 76.70 77.97 75.48 85.19 82.79 97.00 86.83 87.98
 75.41 75.61 52.91 92.96 80.58 84.87 87.94 82.83 67.44 83.19

DATA 43

ORDER

1	2 OF 10.61	4 OF 14.40	4 OF 14.58	
2	4 OF 10.61	4 OF 14.40	4 OF 14.58	
3	4 OF 14.40	4 OF 14.58	2 OF 10.61	
4	1 OF 11.10	1 OF 8.30		
5	3 OF 11.58	2 OF 11.28	1 OF 11.28	
6	1 OF 8.85			
7	3 OF 10.67	1 OF 3.00		
8	1 OF 12.45	1 OF 10.07	1 OF 7.50	
9	6 OF 10.97	8 OF 2.82	3 OF 1.19	
10	1 OF 12.00			
11	3 OF 12.00			
12	8 OF 19.20	4 OF 18.90		
13	1 OF 5.05			
14	8 OF 9.80			
15	18 OF 10.87	16 OF 2.72	13 OF 3.00	
16	2 OF 5.60	1 OF 5.60	1 OF 5.60	
17	2 OF 16.80			
18	2 OF 10.80			
19	2 OF 15.77			
20	10 OF 18.28			
21	1 OF 11.85	1 OF 11.85		
22	1 OF 11.55	1 OF 10.55	1 OF 7.55	1 OF 11.75
23	1 OF 19.14	1 OF 18.27	2 OF 14.57	
24	8 OF 10.54	4 OF 8.33		
25	2 OF 15.25	2 OF 13.72		
26	2 OF 12.19	2 OF 15.24	2 OF 18.28	
27	5 OF 14.63			
28	4 OF 8.95	1 OF 8.86		
29	7 OF 9.00			
30	1 OF 8.31	2 OF 7.89	1 OF 7.39	
31	5 OF 7.49	3 OF 7.87		
32	5 OF 15.25	5 OF 17.07	4 OF 18.30	

FIRST SET OF PARENT LENGTHS

58.27	54.77	56.17	55.11	60.06	57.40	57.81	56.24	60.29	61.32
60.34	54.11	59.58	53.45	56.47	55.76	58.28	55.79	56.24	56.31
57.67	62.21	55.25	59.58	58.24	55.19	56.94	55.90	56.95	56.57
58.83	60.82	60.44	58.30	59.76	57.86	55.32	60.18	61.15	60.30
54.39	59.63	61.27	59.89	56.91	61.44	56.43	60.61	55.84	56.47

SECOND SET OF PARENT LENGTHS

61.90	54.67	55.93	57.51	59.69	61.55	61.06	59.42	56.92	61.73
59.11	61.06	61.34	55.48	52.87	46.83	59.17	60.42	57.36	59.81
47.61	55.30	56.33	57.22	62.32	57.94	62.73	62.92	61.93	58.28
57.24	59.96	54.55	58.62	60.76	60.02	55.23	57.16	57.91	58.95
60.52	56.53	60.51	60.25	61.92	60.23	46.12	60.53	56.41	53.20

DATA 44

ORDER

1	7 OF	10.97	7 OF	12.19	6 OF	13.71	5 OF	15.00
2	22 OF	18.50	35 OF	18.00	12 OF	15.00	56 OF	13.50
	21 OF	12.00						
3	4 OF	15.25						
4	5 OF	13.50	7 OF	12.25	40 OF	15.42		
5	10 OF	13.90	1 OF	11.52	1 OF	17.30	6 OF	16.67
	6 OF	16.27	1 OF	12.67				
6	1 OF	7.80	5 OF	13.50	1 OF	14.80	6 OF	11.00
	4 OF	15.00	5 OF	8.75				
7	3 OF	15.25	7 OF	7.20	2 OF	12.00	14 OF	15.40
	10 OF	12.10	3 OF	12.20				
8	4 OF	9.80	7 OF	17.45	14 OF	8.00	4 OF	12.80
	4 OF	11.00						
9	6 OF	14.80	3 OF	12.20	4 OF	10.05	2 OF	14.55
	2 OF	8.10	1 OF	8.70	1 OF	11.10	4 OF	16.75
	4 OF	17.25	4 OF	17.75	3 OF	12.05	18 OF	15.10
	2 OF	14.90	3 OF	11.80	1 OF	14.50	1 OF	10.20
	8 OF	12.90	9 OF	12.40	2 OF	13.55	2 OF	13.00
	1 OF	12.55	4 OF	16.30				
10	7 OF	11.00	7 OF	12.00	8 OF	14.00	8 OF	15.00
11	2 OF	8.00	5 OF	9.20	2 OF	10.10	23 OF	11.00
	43 OF	12.20	6 OF	13.10	4 OF	14.10	2 OF	15.30
	7 OF	8.00	7 OF	8.15				
12	11 OF	12.00	7 OF	14.00	8 OF	16.00	2 OF	10.00
13	7 OF	12.20	6 OF	14.00	5 OF	15.25	5 OF	16.50
	5 OF	18.30						
14	4 OF	15.00	8 OF	13.50	4 OF	10.00	20 OF	13.25
	2 OF	10.10						
15	9 OF	9.14	7 OF	10.97	25 OF	12.19	20 OF	15.24
16	4 OF	15.30	6 OF	13.70	10 OF	12.20	4 OF	10.70
	30 OF	15.00						
17	9 OF	18.28						
18	18 OF	11.10	6 OF	6.80	2 OF	12.30	6 OF	13.35
19	18 OF	15.00	8 OF	13.00	14 OF	12.50	22 OF	9.20
20	20 OF	15.24						
21	1 OF	18.00	8 OF	15.30	8 OF	14.00	10 OF	12.20
	2 OF	11.00						
22	1 OF	14.25	1 OF	10.30	1 OF	9.24	4 OF	13.15
	7 OF	12.25						
23	40 OF	15.10	26 OF	14.65	10 OF	14.20	7 OF	10.90
24	11 OF	12.20	11 OF	15.85	4 OF	13.71	1 OF	14.63
	20 OF	13.35						
25	32 OF	12.35	12 OF	12.45				
26	51 OF	6.60	11 OF	8.02	10 OF	8.85	17 OF	11.30
	17 OF	12.10	3 OF	13.50				

27	50	OF	18.28								
28	75	OF	18.28								
29	7	OF	16.76	1	OF	9.14					
30	1	OF	13.72	29	OF	12.04	1	OF	12.20		
	5	OF	13.41						1	OF	10.52
31	1	OF	6.55								
32	4	OF	9.15								
33	28	OF	16.76	5	OF	17.52					
34	10	OF	15.24	10	OF	13.71					
35	59	OF	15.24	49	OF	18.28					

FIRST SET OF PARENT LENGTHS

87.03	91.06	76.11	76.06	71.33	76.47	83.83	82.71	77.79	97.35
74.96	83.60	86.97	87.40	83.62	77.09	87.92	77.74	68.14	76.16
86.64	82.39	97.58	81.98	82.60	77.77	75.21	75.36	88.24	83.20
91.01	98.27	83.48	75.43	78.21	77.35	81.21	74.10	78.71	87.36
91.76	70.25	77.68	68.76	65.46	90.85	86.97	90.14	82.09	83.25
90.67	83.75	97.42	83.73	91.59	77.00	91.65	87.94	87.74	86.95
82.00	84.48	82.86	83.82	92.12	82.33	53.92	83.49	83.64	78.39
84.59	87.00	92.66	71.95	80.77	80.03	91.25	76.22	86.07	88.43
78.94	90.80	91.32	82.69	82.25	91.22	91.09	95.49	76.14	81.41
82.13	76.13	95.57	74.18	71.95	82.09	88.93	90.78	81.25	82.49
92.62	88.35	78.46	87.61	97.49	81.44	74.22	83.39	87.34	97.50
77.90	77.94	87.47	82.39	83.09	90.99	88.14	87.89	83.08	91.47
91.06	78.16	81.40	92.92	65.94	80.29	87.29	71.09	86.93	87.76
90.20	53.33	86.16	65.97	67.32	68.19	91.24	71.71	80.04	78.86
83.74	84.69	95.48	84.67	90.68	84.02	66.97	75.64	86.08	80.72
84.56	74.91	87.41	90.20	70.53	98.32	81.11	80.82	90.86	87.81
84.10	76.28	81.77	83.03	82.26	78.25	87.18	92.76	83.93	84.77
87.23	83.42	76.44	77.80	91.80	65.63	75.57	83.69	91.97	83.64
91.08	75.71	92.54	70.84	83.19	78.54	80.53	91.31	88.09	83.75
82.68	83.26	98.46	84.35	86.95	83.58	91.91	87.20	91.02	83.29
77.55	95.62	65.79	82.09	75.45	71.82	81.87	90.82	82.14	83.39
88.09	81.99	77.13	77.86	75.00	91.24	95.39	87.16	91.39	83.90
95.88	76.11	78.74	87.44	90.99	77.95	82.75	87.98	91.11	83.82
82.86	74.81	75.10	87.27	68.71	82.86	78.76	82.75	82.67	77.24
84.39	88.14	83.32	88.69	83.24	91.18	76.97	91.12	92.03	87.07
84.12	83.10	87.47	80.87	92.04	87.35	91.70	77.08	82.17	91.24
53.87	65.09	66.69	53.30	90.57	91.70	83.00	90.74	78.37	84.60
92.22	80.88	75.31	84.92	97.67	67.74	80.40	81.67	83.42	78.48
75.10	76.27	98.72	90.90	88.86	83.06	77.58	88.98	88.60	92.76
82.20	91.34	76.18	83.03	82.51	83.77	75.98	87.92	75.66	86.71
98.30	87.36	80.51	86.75	53.92	75.74	91.14	82.15	82.61	94.31
80.36	81.31	77.59	83.76	48.19	91.30	91.06	87.65	91.37	91.33
77.67	82.98	77.86	75.30	72.05	78.64	83.33	87.22	87.31	83.92

SECOND SET OF PARENT LENGTHS

82.68	83.26	98.46	84.35	86.95	83.58	91.91	87.20	91.02	83.29
77.55	95.62	65.79	82.09	75.45	71.82	81.87	90.82	82.14	83.39
88.09	81.99	77.13	77.86	75.00	91.24	95.39	87.16	91.39	83.90
95.88	76.11	78.74	87.44	90.99	77.95	82.75	87.98	91.11	83.82
82.86	74.81	75.10	87.27	68.71	82.86	78.76	82.75	82.67	77.24
84.39	88.14	83.32	88.69	83.24	91.18	76.97	91.12	92.03	87.07
84.12	83.10	87.47	80.87	92.04	87.35	91.70	77.08	82.17	91.24
53.87	65.09	66.69	53.30	90.57	91.70	83.00	90.74	78.37	84.60
92.22	80.88	75.31	84.92	97.67	67.74	80.40	81.67	83.42	78.48
75.10	76.27	98.72	90.90	88.86	83.06	77.58	88.98	88.60	92.76
82.20	91.34	76.18	83.03	82.51	83.77	75.98	87.92	75.66	86.71
98.30	87.36	80.51	86.75	53.92	75.74	91.14	82.15	82.61	94.31
80.36	81.31	77.59	83.76	48.19	91.30	91.06	87.65	91.37	91.33
77.67	82.98	77.86	75.30	72.05	78.64	83.33	87.22	87.31	83.92
81.96	77.93	91.88	91.94	94.66	91.52	83.16	84.28	91.20	48.64
92.03	83.39	81.02	78.59	82.37	91.91	80.07	88.25	94.85	91.83
81.35	83.60	66.41	86.01	84.40	90.23	91.78	91.63	80.75	83.79
72.98	59.72	82.51	91.56	87.78	88.64	86.78	92.89	87.28	80.63
75.23	95.74	88.31	94.23	95.53	77.14	81.02	77.85	84.90	82.74

76.32	83.24	48.58	83.27	72.40	92.00	72.34	80.05	80.25	81.05
84.99	82.51	84.13	83.17	71.87	84.66	97.07	83.50	83.35	90.60
82.40	80.99	74.33	92.04	87.22	87.96	77.74	91.77	81.93	77.57
87.05	77.19	77.67	84.30	84.74	77.77	75.90	70.50	91.85	87.58
84.86	91.86	67.42	92.81	92.04	84.90	77.06	77.21	87.74	84.51
71.37	77.64	87.53	78.38	60.51	86.56	92.77	83.60	78.65	60.50
90.09	88.06	78.52	84.60	83.90	76.00	77.85	80.10	83.91	76.53
77.93	87.83	86.59	75.07	98.05	87.70	80.70	94.90	81.06	80.23
77.79	97.66	81.83	98.02	95.67	94.80	75.75	92.28	87.95	87.13
83.25	82.30	65.51	82.32	77.31	82.97	94.02	91.35	81.91	87.28
82.43	92.08	80.58	77.79	95.46	59.68	86.88	87.18	77.13	80.18
82.89	91.71	86.22	83.96	84.73	87.74	80.82	75.24	83.06	82.22
80.76	83.57	90.55	90.19	77.19	95.36	91.42	82.31	75.02	83.35
75.91	92.28	74.45	95.15	83.80	87.45	87.46	77.68	77.90	83.24

DATA 45

ORDER

1	24 OF 11.10	32 OF 13.25	32 OF 12.64	32 OF 12.95
	1 OF 12.95	1 OF 12.95		
2	11 OF 12.80	1 OF 12.35	1 OF 6.40	
3	9 OF 15.25	8 OF 16.77	8 OF 18.30	
4	55 OF 18.30	27 OF 17.00	25 OF 14.00	14 OF 15.30
	17 OF 12.20			
5	1 OF 14.05	3 OF 15.25	3 OF 18.30	5 OF 11.00
	11 OF 12.82			
6	39 OF 13.65			
7	10 OF 10.70	3 OF 12.20	7 OF 11.00	
8	9 OF 14.00	13 OF 15.00	2 OF 18.41	
9	7 OF 10.97	6 OF 12.19	9 OF 15.24	
10	7 OF 12.00	3 OF 14.00	4 OF 16.00	2 OF 9.00
	3 OF 10.00	7 OF 11.00	2 OF 13.00	
11	5 OF 14.00	5 OF 13.00	5 OF 12.00	4 OF 11.00
	4 OF 10.00	4 OF 9.00	4 OF 8.00	
12	1 OF 18.00	6 OF 15.30	6 OF 14.00	6 OF 12.20
	7 OF 11.00			
13	16 OF 10.97	12 OF 12.19	10 OF 13.71	9 OF 15.24
	9 OF 16.76	8 OF 18.28		
14	3 OF 7.70	2 OF 9.30	1 OF 9.70	28 OF 15.35
	6 OF 15.60			
15	5 OF 8.55	3 OF 11.20	4 OF 18.45	2 OF 12.80
	1 OF 12.50	1 OF 16.00		
16	1 OF 12.30	3 OF 11.80	1 OF 6.00	3 OF 12.30
	8 OF 17.25	3 OF 11.35	3 OF 7.90	1 OF 6.20
17	2 OF 15.25			
18	26 OF 15.24	22 OF 18.28		
19	10 OF 18.28	10 OF 13.71		
20	3 OF 7.89			
21	6 OF 16.76	10 OF 17.52	1 OF 9.14	10 OF 18.28
22	23 OF 18.28			
23	60 OF 18.28			
24	23 OF 15.24	19 OF 18.28		
25	10 OF 18.28	10 OF 16.76	10 OF 15.24	

FIRST SET OF PARENT LENGTHS

68.27	92.87	63.75	64.67	84.29	91.67	68.40	87.38	68.66	83.52
80.22	72.59	68.56	69.97	71.82	64.15	65.46	68.44	73.48	79.87
92.90	65.50	69.90	72.89	68.18	71.10	87.96	75.80	60.18	69.85
76.44	63.01	77.06	68.23	64.83	92.92	77.02	76.87	92.00	69.18
72.08	69.79	69.02	68.00	75.78	68.66	68.88	73.32	72.99	87.02
71.22	69.11	84.59	76.58	72.13	47.59	68.28	71.40	69.86	76.51
88.31	80.22	64.57	85.39	85.20	84.65	73.11	71.76	75.55	71.45
72.74	68.40	72.73	72.75	92.90	72.64	64.68	87.35	69.95	75.53
77.62	84.92	92.92	72.18	92.83	68.28	75.25	72.95	76.36	72.65

93.59 92.69 72.85 63.85 71.46 72.06 72.20 80.65 68.09 69.70
 69.37 75.91 93.12 68.53 69.12 81.90 76.35 71.99 81.75 76.54
 69.53 77.28 85.86 69.64 69.06 65.61 72.10 75.12 73.76 89.47
 88.92 71.33 64.64 71.85 68.31 75.24 87.61 92.51 69.55 69.71
 83.27 92.23 75.98 73.74 81.63 68.07 72.73 65.74 87.88 64.57
 93.48 72.74 92.13 87.06 72.20 71.67 87.15 88.88 85.93 76.59
 79.16 76.65 52.47 83.95 69.51 76.47 69.19 72.87 72.52 88.24
 63.76 76.33 93.15 87.87 68.88 69.46 68.78 80.52 76.07 92.78
 73.98 92.87 85.38 87.44 68.23 76.38 75.15 76.51 76.69 89.51
 85.79 84.18 60.94 87.99 81.49 80.02 60.71 68.07 93.99 92.27
 93.70 71.77 77.32 69.02 84.22 64.10 87.58 68.55 68.06 84.37
 85.65 76.50 68.15 72.39 87.95 93.15 80.35 92.73 84.21 65.68

SECOND SET OF PARENT LENGTHS

93.70 71.77 77.32 69.02 84.22 64.10 87.58 68.55 68.06 84.37
 85.65 76.50 68.15 72.39 87.95 93.15 80.35 92.73 84.21 65.68
 88.72 64.12 92.24 92.32 71.70 65.32 88.59 69.61 88.34 72.47
 72.77 76.40 85.44 84.02 81.17 92.84 88.53 85.55 76.51 73.12
 64.09 88.49 87.09 83.10 88.81 85.89 69.03 73.19 93.81 88.14
 72.55 93.98 72.93 85.77 92.16 53.08 72.97 72.12 53.99 84.81
 80.91 84.20 84.97 85.99 73.21 88.33 88.11 75.67 83.00 69.97
 84.77 84.89 72.40 75.41 77.86 93.41 85.71 80.59 88.13 75.48
 69.69 72.77 89.42 68.60 68.79 72.34 75.88 84.23 76.44 84.54
 80.25 85.59 77.26 76.24 64.09 77.36 89.31 69.64 87.04 73.46
 72.37 72.07 61.07 83.81 61.16 85.71 73.74 76.04 72.63 83.34
 63.40 53.30 83.14 93.14 85.53 83.93 77.79 72.34 89.90 84.29
 84.62 73.09 63.87 88.46 87.88 71.09 72.64 85.00 71.24 75.45
 87.46 72.71 71.13 87.35 84.93 88.38 77.89 76.87 79.23 68.52
 69.07 81.66 91.35 85.15 88.68 73.75 69.38 64.48 84.44 88.28
 72.72 61.76 73.01 76.25 71.36 88.92 83.26 88.25 69.11 92.42
 63.51 77.25 63.86 69.93 76.79 81.32 69.84 69.11 68.06 72.41
 73.84 73.34 93.53 72.04 87.48 73.52 88.80 76.12 76.47 68.75
 93.23 72.66 52.84 69.12 88.11 84.53 89.21 72.47 73.92 61.21
 75.01 61.12 68.61 69.55 88.76 72.61 73.84 73.48 75.30 67.49
 68.20 71.81 93.06 69.00 71.50 72.97 93.28 85.92 63.00 64.73

DATA 46

ORDER

1	5 OF	15.25	5 OF	16.77	5 OF	18.30		
2	22 OF	18.50	12 OF	15.35	26 OF	14.00		
3	2 OF	12.04	6 OF	12.04	3 OF	12.04	4 OF	12.04
	1 OF	6.00						
4	2 OF	8.00	3 OF	9.20	5 OF	10.10	4 OF	11.00
	9 OF	7.50	5 OF	12.20	2 OF	13.10	4 OF	14.10
	6 OF	15.30						
5	9 OF	14.00	13 OF	15.00				
6	7 OF	18.28						
7	2 OF	9.20	4 OF	15.30	6 OF	13.70	8 OF	12.20
	5 OF	10.70						
8	21 OF	9.14	29 OF	15.24				
9	24 OF	9.20	14 OF	15.60	3 OF	12.75	2 OF	12.20
	2 OF	11.50						
10	1 OF	9.07						
11	1 OF	7.31						
12	2 OF	7.62						
13	10 OF	18.28	10 OF	13.71				
14	35 OF	18.28						
15	14 OF	18.28						

FIRST SET OF PARENT LENGTHS

80.31	53.15	63.12	75.33	80.91	77.47	68.57	64.19	78.98	70.16
78.15	77.43	77.21	65.36	82.24	77.21	67.04	80.34	78.70	81.06
83.11	66.05	71.33	77.46	81.79	82.59	74.42	81.25	77.65	70.69
71.24	83.20	79.02	82.28	80.50	81.49	79.40	68.91	80.91	74.20
73.03	81.32	82.61	74.82	65.38	72.86	74.75	65.74	61.62	67.09
80.92	77.74	64.15	79.18	82.69	65.56	80.08	81.48	82.10	79.28
80.76	66.00	81.15	74.94	82.22	80.89	65.32	80.94	79.74	72.96
83.05	68.64	64.32	82.20	73.26	82.75	78.16	80.22	75.82	68.93

SECOND SET OF PARENT LENGTHS

71.83	82.84	62.52	61.52	79.42	69.83	64.24	80.93	81.41	78.78
81.73	80.66	67.33	69.09	81.50	65.21	77.75	81.59	66.76	82.23
81.57	69.31	79.67	79.22	57.29	79.70	77.61	82.28	80.21	65.71
82.39	82.87	67.73	79.51	64.53	68.56	65.56	58.26	67.53	68.80
65.00	79.85	81.05	78.61	82.19	76.66	73.23	82.40	80.32	81.31
67.91	66.69	78.93	76.38	61.87	66.84	77.12	82.21	72.12	79.82
79.84	68.62	67.09	83.00	72.10	81.65	58.94	83.79	74.25	78.34
81.77	67.53	57.21	64.46	76.89	77.12	78.73	76.27	69.04	79.84

DATA 47

ORDER

1	20 OF 18.30	5 OF 16.00	10 OF 15.30	7 OF 14.00
	5 OF 12.60	5 OF 12.00		
2	10 OF 10.75			
3	5 OF 9.15	5 OF 9.75	8 OF 10.97	7 OF 12.19
	9 OF 13.71	6 OF 15.00		
4	1 OF 7.85	2 OF 8.05	1 OF 12.05	1 OF 12.10
	1 OF 9.30	1 OF 10.30	1 OF 12.00	1 OF 9.20
	1 OF 13.82	1 OF 12.54	1 OF 12.56	1 OF 12.58
	1 OF 12.94	1 OF 10.12		
5	12 OF 10.70	1 OF 7.40		
6	1 OF 20.73	1 OF 20.73		
7	3 OF 8.45	6 OF 12.20		
8	7 OF 11.70	7 OF 9.70	16 OF 13.20	5 OF 8.85
9	6 OF 7.37			
10	3 OF 15.25	5 OF 16.47	3 OF 18.30	
11	2 OF 13.80	1 OF 6.10	10 OF 14.10	
12	9 OF 12.00	6 OF 16.00	4 OF 9.45	12 OF 12.00
	1 OF 13.22	2 OF 11.52	4 OF 11.55	
13	3 OF 10.00	4 OF 15.00		
14	3 OF 15.30	5 OF 9.80	4 OF 11.00	4 OF 12.20
	3 OF 14.10			
15	6 OF 15.30			
16	8 OF 13.80			
17	2 OF 22.25			
18	8 OF 11.00	7 OF 13.00	7 OF 15.00	7 OF 18.00
19	4 OF 14.00	2 OF 11.00	1 OF 8.55	
20	1 OF 12.39	2 OF 11.83	1 OF 11.37	
21	7 OF 15.24	6 OF 18.28		
22	1 OF 11.12	2 OF 10.85	1 OF 10.10	1 OF 9.45
	1 OF 13.60	2 OF 8.17	2 OF 8.10	1 OF 12.17
23	1 OF 16.55	1 OF 9.81	1 OF 13.60	2 OF 8.15
	1 OF 6.10			
24	2 OF 9.50	7 OF 10.00		
25	3 OF 12.25	3 OF 9.25	4 OF 10.45	1 OF 14.75
	1 OF 8.85			
26	1 OF 11.62	1 OF 11.20		
27	2 OF 11.70	6 OF 8.47		
28	4 OF 6.57			
29	10 OF 15.24	12 OF 18.28		
30	36 OF 8.61	1 OF 11.60		

31	5 OF	16.76							
32	15 OF	12.19	10 OF	18.28					
33	2 OF	14.92	4 OF	9.69	1 OF	9.64	14 OF	9.36	
	12 OF	9.32							
34	1 OF	8.47							
35	30 OF	8.68	1 OF	8.58	68 OF	8.50	8 OF	8.38	
	88 OF	8.33	7 OF	8.28	9 OF	8.10	1 OF	9.44	
36	10 OF	8.48	7 OF	8.40	3 OF	8.30	3 OF	6.65	

FIRST SET OF PARENT LENGTHS

92.78	73.62	80.72	79.71	87.75	88.03	88.44	86.35	83.16	87.78
89.19	84.12	93.98	81.72	84.52	80.60	84.92	85.04	87.97	85.46
78.99	80.76	90.64	88.96	88.99	90.26	83.64	77.46	92.99	81.76
87.62	89.91	84.84	85.81	93.27	82.36	80.67	83.82	86.80	76.44
87.41	86.53	83.45	78.88	79.25	87.55	91.99	82.02	71.17	76.80
83.24	84.27	89.42	83.06	85.54	89.65	65.06	85.46	83.24	90.26
81.39	79.97	83.26	92.81	90.53	88.89	83.52	85.09	74.86	90.38

77.48	84.48	87.55	87.93	86.65	85.48	84.03	87.90	89.04	74.15
80.55	84.95	85.71	89.67	86.61	87.63	86.25	76.80	86.51	86.81
85.29	85.40	85.80	90.20	83.96	90.92	86.86	92.87	72.45	84.90
86.31	88.71	87.48	80.49	91.56	84.98	89.79	80.91	84.33	86.77
86.65	89.99	82.07	78.51	84.37	91.65	92.56	86.48	86.79	92.41
83.34	86.31	86.78	89.87	79.22	87.46	79.73	76.24	80.82	86.74

SECOND SET OF PARENT LENGTHS

93.98	79.29	83.97	84.16	86.19	88.70	87.44	85.32	85.91	90.60
87.37	81.86	84.80	82.03	84.91	86.23	81.17	84.87	86.69	92.31
90.65	80.08	86.56	85.69	90.02	81.93	83.38	89.89	84.89	92.30
77.73	89.68	79.50	89.86	80.66	78.24	87.42	79.39	89.49	86.47
91.37	87.97	76.49	84.23	89.89	82.28	86.69	65.56	82.15	85.83
79.58	83.99	90.69	83.19	78.91	80.72	92.13	85.27	64.39	86.92
82.98	81.55	91.49	83.95	64.33	82.35	87.13	83.64	86.65	90.06
83.51	80.52	87.51	78.39	83.77	84.04	91.35	86.66	87.81	79.91
89.20	80.97	93.00	91.25	86.48	87.54	75.97	93.91	84.68	92.91
85.30	80.66	86.18	84.17	87.34	84.50	79.97	90.26	83.88	86.89
80.37	86.22	77.99	85.96	88.83	84.37	84.69	86.82	85.67	85.64
85.76	84.78	85.81	86.79	72.45	71.60	90.51	84.64	80.25	80.73
80.16	64.34	86.62	84.60	85.01	90.65	90.77	80.82	85.91	88.11

DATA 48

ORDER

1	200 OF	14.60							
2	4 OF	7.46							
3	1 OF	18.05	2 OF	12.05					
4	1 OF	9.20							
5	4 OF	8.27							
6	1 OF	8.00							
7	1 OF	16.00							
8	7 OF	10.10							
9	6 OF	11.00	6 OF	13.00	6 OF	15.00	6 OF	18.00	
10	12 OF	14.00							
11	1 OF	11.97							
12	3 OF	14.05							
13	1 OF	13.35							
14	2 OF	14.60							
15	2 OF	13.52	1 OF	8.87	1 OF	13.30	1 OF	12.00	
16	1 OF	13.50	1 OF	6.02					
17	2 OF	12.10							
18	1 OF	13.50							
19	4 OF	15.24	6 OF	18.28					
20	56 OF	9.85	5 OF	9.80	1 OF	8.68			

FIRST SET OF PARENT LENGTHS

72.31	65.93	76.76	70.14	66.01	75.76	66.50	75.11	66.17	74.99
65.44	73.70	74.17	74.76	78.99	59.10	73.64	71.90	74.66	72.86
72.15	68.20	66.82	74.07	76.05	76.66	79.44	64.69	76.16	67.72
79.87	62.72	74.51	74.53	66.55	74.52	69.15	76.19	76.79	74.00
77.84	77.02	68.52	76.92	76.84	73.76	76.99	77.19	65.63	73.88
73.61	74.79	67.69	76.80	75.61	65.46	76.21	70.13	71.86	70.94
70.90	79.95	72.55	65.78	75.68	65.02	76.03	77.95	77.45	75.03
68.46	77.01	75.86	73.06	75.65	65.32	75.03	64.34	76.74	66.33

SECOND SET OF PARENT LENGTHS

73.85	75.21	66.61	65.74	77.95	77.01	69.52	73.97	76.12	65.03
62.07	74.13	64.12	61.52	64.04	71.54	64.85	75.21	66.60	66.71
78.87	63.78	76.83	75.90	72.91	59.41	75.21	75.54	76.35	76.22
76.14	73.83	70.72	70.81	64.41	64.16	75.20	76.75	77.74	68.64
69.14	75.08	77.23	64.61	65.55	65.81	69.90	76.12	74.59	77.40
76.17	74.44	65.05	73.30	76.38	65.56	77.92	75.43	74.26	74.70
75.83	76.66	73.45	71.78	64.56	75.34	75.01	74.94	63.54	71.80
74.88	78.05	75.38	77.82	68.43	59.20	77.33	73.15	69.92	76.19

DATA 49

ORDER

1	4 OF 14.00			
2	1 OF 17.21	1 OF 17.22	1 OF 17.22	
3	8 OF 13.15			
4	2 OF 14.05			
5	1 OF 8.00			
6	6 OF 9.15	7 OF 11.00	5 OF 13.00	2 OF 14.00
7	2 OF 8.05	1 OF 13.42	1 OF 11.32	1 OF 9.52
8	1 OF 14.90	1 OF 13.42	2 OF 8.02	
9	2 OF 12.73			
10	2 OF 13.50	3 OF 13.20	1 OF 11.67	

FIRST SET OF PARENT LENGTHS

68.94	66.71	67.76	58.16	67.13	67.26	70.41	59.12	67.98	67.84
71.14	58.30	70.54	54.51	67.16	70.35	67.70	65.97	69.55	64.53

SECOND SET OF PARENT LENGTHS

65.05	67.28	66.23	70.83	66.87	66.73	58.58	69.87	66.01	66.16
54.85	70.69	58.45	71.48	65.83	55.64	65.29	67.02	59.44	69.46

DATA 50

ORDER

1	9 OF 12.50	3 OF 13.50	2 OF 16.27	6 OF 8.00
	6 OF 9.00	6 OF 10.00	6 OF 11.00	
2	1 OF 9.05	2 OF 12.35		
3	1 OF 13.30			
4	1 OF 5.92			
5	1 OF 10.90			
6	7 OF 8.85	6 OF 9.50	1 OF 8.00	
7	2 OF 11.02			
8	1 OF 14.05	1 OF 12.30		
9	2 OF 17.62	2 OF 17.52	2 OF 9.30	2 OF 17.22
	2 OF 13.94	2 OF 17.32		
10	2 OF 11.00	3 OF 13.00	16 OF 6.75	14 OF 6.60
	4 OF 6.90	3 OF 9.80		
11	1 OF 9.00			
12	1 OF 12.00			
13	1 OF 12.62			
14	6 OF 9.00	6 OF 10.00	10 OF 11.00	9 OF 12.00
	8 OF 14.00	7 OF 16.00		

15	6 OF	7.95							
16	3 OF	10.40							
17	1 OF	12.20	1 OF	16.77					
18	1 OF	7.90							
19	14 OF	8.97	2 OF	11.92	17 OF	10.25	7 OF	11.20	
	4 OF	9.55	10 OF	9.42					
20	92 OF	6.47	66 OF	9.22	90 OF	12.65	24 OF	13.17	
	1 OF	11.65	6 OF	8.30					
21	1 OF	7.15	2 OF	10.87	4 OF	12.87	2 OF	9.75	
	1 OF	10.15	2 OF	7.15					
22	4 OF	6.77							
23	1 OF	6.35							
24	6 OF	18.28							
25	8 OF	12.00	23 OF	6.10	1 OF	7.05			
26	1 OF	10.07							
27	1 OF	4.90							
28	3 OF	10.72							
29	5 OF	13.71	3 OF	12.19	4 OF	9.14	4 OF	7.62	
	3 OF	13.72	4 OF	15.25					

FIRST SET OF PARENT LENGTHS

86.11	83.46	92.78	81.46	89.78	90.46	91.76	88.42	83.66	94.10
88.69	89.20	82.98	83.07	80.59	87.90	83.09	90.42	76.63	74.03
82.48	84.57	85.11	82.45	85.74	82.35	89.42	93.40	90.59	90.91
92.12	89.50	92.90	86.83	86.87	84.75	80.63	97.02	85.46	84.50
88.89	76.80	79.79	77.51	86.90	88.84	83.92	97.88	87.02	92.20
86.99	82.11	92.73	95.31	82.31	76.08	92.61	86.97	83.29	86.01
85.41	90.36	89.46	85.53	94.97	91.04	86.50	91.63	87.27	84.93
86.18	86.67	81.40	86.35	87.43	77.41	84.61	85.96	89.27	80.95
81.23	86.83	86.89	83.79	94.75	82.39	82.55	92.79	80.77	85.44
80.69	76.21	85.01	88.15	85.84	85.65	88.32	80.09	84.66	83.10

SECOND SET OF PARENT LENGTHS

87.10	89.38	86.34	88.58	80.44	87.39	81.36	88.67	86.70	88.16
83.67	79.60	91.49	87.53	95.79	56.93	86.49	82.53	83.78	90.91
90.43	85.37	86.30	86.48	87.13	87.45	94.56	97.25	82.43	97.37
85.32	82.53	76.28	83.91	83.92	80.32	90.65	85.01	89.24	74.30
86.59	88.85	82.76	89.84	87.23	86.76	74.48	82.01	76.73	83.26
83.99	80.55	89.34	83.08	86.44	83.88	79.35	86.11	88.52	92.14
93.12	88.44	86.52	87.13	94.09	89.35	89.26	82.35	80.75	88.31
80.14	86.98	90.63	85.97	88.11	92.89	95.36	95.16	79.69	87.64
74.62	81.92	97.97	77.48	86.13	89.46	85.59	81.34	98.71	80.19
89.76	83.84	82.20	89.68	88.24	93.28	97.55	77.72	88.37	97.68

DATA 51

ORDER

1	2 OF	10.05	22 OF	13.60	4 OF	14.87	1 OF	10.95	
	1 OF	9.10							
2	28 OF	10.15	7 OF	13.75	7 OF	15.60	7 OF	18.30	
3	8 OF	14.00							
4	20 OF	21.95	1 OF	9.05	1 OF	20.19	1 OF	17.77	
	5 OF	12.05	2 OF	8.90	2 OF	9.00			
5	2 OF	9.20	2 OF	12.20	2 OF	15.30	2 OF	11.00	
	2 OF	13.10	2 OF	14.10	1 OF	12.20	6 OF	11.10	
	12 OF	9.30	6 OF	10.15					
6	14 OF	18.10	14 OF	9.00					
7	2 OF	9.14	24 OF	14.59	12 OF	11.64	12 OF	11.94	
	6 OF	8.38							
8	22 OF	15.25	1 OF	12.20					
9	5 OF	12.00	8 OF	14.00	7 OF	16.00			
10	5 OF	10.97	5 OF	12.19	4 OF	14.02	7 OF	15.24	
11	8 OF	21.75	2 OF	21.90					
12	10 OF	14.00	8 OF	13.40	10 OF	12.20			
13	12 OF	18.20	16 OF	17.10	16 OF	10.10	8 OF	9.70	

14	1 OF 6.25	2 OF 13.75	2 OF 12.45	2 OF 11.90
	30 OF 11.10			
15	52 OF 20.15			
16	8 OF 13.40	10 OF 15.24		
17	2 OF 20.75			
18	5 OF 15.00	6 OF 9.15	6 OF 11.00	5 OF 12.20
	5 OF 14.00			
19	21 OF 11.50			
20	7 OF 16.75			
21	2 OF 12.80	3 OF 12.20	6 OF 11.00	2 OF 15.30
	3 OF 14.70	2 OF 13.40	2 OF 14.00	
22	7 OF 18.25	6 OF 18.45		
23	3 OF 8.60	12 OF 9.20		
24	6 OF 10.25	3 OF 9.05	1 OF 11.00	1 OF 4.90
25	11 OF 12.04	1 OF 10.52	1 OF 9.15	1 OF 6.71
26	1 OF 8.70	2 OF 11.10	1 OF 5.85	1 OF 6.85
	2 OF 7.55	2 OF 7.90		
27	4 OF 13.12	3 OF 14.35	1 OF 17.70	1 OF 16.47
	1 OF 11.00	2 OF 15.85		
28	2 OF 10.67	3 OF 6.10		
29	1 OF 16.76			
30	50 OF 18.28			
31	33 OF 15.24	28 OF 18.28		
32	15 OF 18.28			
33	24 OF 15.24			
34	11 OF 9.37	4 OF 8.87	1 OF 10.05	1 OF 8.28
35	1 OF 8.26	2 OF 7.87	1 OF 7.68	1 OF 7.67
	1 OF 7.65	1 OF 7.48	1 OF 7.64	
36	1 OF 15.25	4 OF 17.07	3 OF 18.30	3 OF 13.12
37	44 OF 12.50			

FIRST SET OF PARENT LENGTHS

61.99	81.96	84.84	81.38	87.73	68.93	89.00	82.61	84.67	77.49
90.94	79.18	85.62	80.09	87.99	58.10	83.63	87.89	83.62	82.72
86.74	81.97	80.11	85.96	81.72	61.65	65.40	87.58	80.81	82.68
75.76	79.39	86.51	87.54	87.57	81.57	85.29	82.62	60.07	83.84
89.35	72.56	84.13	83.77	86.39	83.42	84.96	81.00	81.33	80.97
83.88	86.49	90.07	84.95	81.06	74.81	79.26	80.28	86.32	82.32
84.04	84.36	84.79	86.47	82.75	61.26	80.73	82.06	72.76	84.02
83.27	79.43	74.13	82.89	68.14	82.78	82.44	82.57	81.42	81.38
60.49	90.45	83.32	74.82	83.05	82.89	86.88	84.27	84.65	83.44
69.79	57.73	82.29	76.12	69.06	86.29	87.21	71.62	82.83	65.35
85.63	87.63	81.09	86.89	85.50	84.00	84.45	74.73	82.27	80.06
85.88	86.76	74.61	80.83	82.44	81.20	81.18	70.29	82.08	83.85
83.37	82.62	81.32	57.33	83.07	76.45	65.08	86.35	83.38	80.13
80.36	80.94	85.40	83.94	81.03	82.71	86.96	87.37	76.58	86.10
81.38	79.37	85.83	84.62	81.20	80.60	84.75	47.13	79.97	84.64
75.14	80.03	85.94	87.32	83.41	83.60	82.88	87.88	80.34	86.14
83.76	68.29	81.82	83.33	83.57	84.46	60.59	84.39	70.01	81.52
81.02	84.39	76.26	83.02	87.79	69.47	85.76	81.26	64.75	76.10
86.90	80.45	81.57	90.34	86.85	76.03	82.56	76.04	69.21	84.88
71.20	84.22	84.51	83.03	85.55	81.01	84.12	82.59	85.48	86.55

SECOND SET OF PARENT LENGTHS

71.20	84.22	84.51	83.03	85.55	81.01	84.12	82.59	85.48	86.55
89.00	83.03	81.15	84.61	69.26	87.06	54.99	83.38	80.32	85.50
60.05	85.81	79.37	85.90	69.00	89.89	81.36	69.11	82.37	82.27
75.25	83.02	84.88	75.03	83.27	86.34	86.59	68.42	84.18	82.31
86.23	85.60	75.48	71.46	71.42	83.42	80.70	83.37	90.92	81.15
58.64	86.43	80.86	82.22	75.60	81.58	80.03	84.99	83.67	84.02
82.12	72.50	60.92	80.04	84.93	87.13	85.73	84.71	74.67	83.67
80.95	81.63	80.21	76.52	82.24	86.74	85.26	82.93	86.23	81.97
82.72	85.56	86.86	82.10	87.85	83.21	82.55	82.42	84.57	83.61
87.51	53.54	81.67	87.17	82.94	83.10	75.11	81.72	80.34	81.55
86.20	90.26	82.70	85.88	87.93	70.70	71.78	87.37	83.16	86.64
75.36	74.36	83.90	76.10	75.49	81.99	81.54	86.26	83.72	84.94

75.11 75.23 87.38 84.16 83.55 84.79 84.81 86.70 82.92 82.15
 82.62 83.37 84.67 90.66 81.92 85.54 86.91 75.64 81.61 84.96
 84.63 84.05 79.59 82.05 83.96 83.28 70.93 71.62 85.41 76.89
 84.61 85.62 77.16 81.37 83.79 84.39 80.24 89.86 85.02 81.35
 86.85 85.96 79.05 74.68 82.58 81.39 92.11 69.11 84.65 76.85
 81.23 87.71 83.17 82.66 82.42 80.53 87.40 81.60 86.98 83.42
 83.98 80.60 86.73 82.97 68.21 87.52 76.24 83.73 87.25 86.89
 74.09 84.54 83.42 57.65 75.14 85.96 82.43 86.95 87.72 80.11

DATA 52

ORDER

1	12 OF	15.60							
2	38 OF	18.30	33 OF	15.50	20 OF	13.00			
3	12 OF	15.60							
4	19 OF	12.65	8 OF	12.05					
5	1 OF	11.81							
6	1 OF	19.00	11 OF	17.07	6 OF	15.00	7 OF	14.00	
	8 OF	12.00							
7	2 OF	16.77	1 OF	16.17	1 OF	12.50	1 OF	11.60	
8	24 OF	14.65	50 OF	13.80					
9	7 OF	14.00							
10	20 OF	9.15							
11	5 OF	15.00	3 OF	9.00					
12	2 OF	16.00	6 OF	12.00	5 OF	11.00	2 OF	10.00	
	5 OF	9.00							
13	3 OF	15.00	4 OF	12.00	3 OF	9.00			
14	10 OF	17.75	2 OF	17.62	2 OF	17.52	2 OF	17.42	
	2 OF	17.32	2 OF	17.22	1 OF	14.72			
15	8 OF	11.25							
16	2 OF	23.75	3 OF	15.00	2 OF	10.00			
17	1 OF	18.30	4 OF	13.00	2 OF	12.00	2 OF	11.50	
	10 OF	11.25	4 OF	11.00					
18	1 OF	15.30	1 OF	14.70	1 OF	14.00	2 OF	13.40	
	2 OF	12.80	1 OF	12.20	4 OF	11.00			
19	15 OF	15.24							
20	5 OF	18.50	6 OF	15.50					
21	5 OF	14.50	10 OF	13.75	5 OF	13.10	5 OF	11.90	
22	5 OF	18.29	6 OF	15.24	19 OF	12.19			
23	2 OF	11.00	19 OF	10.60					
24	10 OF	18.40	1 OF	13.50					
25	21 OF	11.30							
26	218 OF	7.90							
27	2 OF	12.20	2 OF	11.80	2 OF	7.05			
28	1 OF	10.97							
29	1 OF	10.10							
30	35 OF	18.28							
31	1 OF	9.91	2 OF	8.28	1 OF	8.28	9 OF	7.89	
	5 OF	7.89							
32	1 OF	7.51	1 OF	7.50	1 OF	8.38	1 OF	8.26	
	1 OF	7.46	2 OF	7.46	1 OF	8.26	1 OF	8.20	
	2 OF	7.65	3 OF	7.48	9 OF	7.48	3 OF	7.47	
	2 OF	7.50	2 OF	7.51	4 OF	7.87	1 OF	7.68	
33	36 OF	9.15	2 OF	9.15					

FIRST SET OF PARENT LENGTHS

71.87 74.24 72.56 69.18 73.02 75.42 75.35 74.29 74.59 76.88
 71.98 67.92 69.72 75.03 75.66 75.73 63.41 78.84 75.37 79.66
 66.26 78.15 72.53 75.84 68.24 67.83 74.83 71.48 74.41 69.11
 75.92 69.50 78.72 58.79 75.20 63.08 78.71 75.48 75.54 62.85
 72.26 70.91 74.05 76.15 70.41 76.08 67.84 77.26 69.00 74.63
 58.73 73.69 74.54 75.99 77.05 76.38 72.85 81.66 71.33 68.97
 67.88 74.51 71.11 66.08 71.44 72.94 75.67 68.50 69.95 75.22
 71.75 75.49 74.18 78.67 74.34 74.02 74.02 67.94 78.42 69.00

72.22 78.28 75.69 78.55 87.13 78.75 73.33 64.25 64.45 74.49
 72.79 75.36 78.03 64.96 69.45 75.11 72.53 76.21 70.48 75.93
 69.26 74.13 68.49 68.71 57.85 62.67 62.35 75.04 71.08 75.10
 76.90 68.45 64.56 79.30 64.73 73.69 67.52 76.94 72.90 73.95
 74.56 72.79 65.71 72.13 75.36 72.95 69.45 75.09 69.49 76.10
 68.13 69.89 74.12 77.72 78.26 75.02 71.76 78.41 73.57 73.70
 75.07 64.07 74.79 75.10 68.52 66.13 72.37 74.59 72.23 75.05
 69.25 74.98 66.66 75.09 74.60 70.81 69.38 80.00 81.60 46.54

SECOND SET OF PARENT LENGTHS

73.12 71.75 72.43 75.81 71.97 69.57 70.64 69.70 71.40 68.11
 73.02 76.07 75.27 69.97 68.33 69.26 77.58 64.15 68.62 57.39
 77.73 64.84 72.46 69.15 75.75 76.16 70.16 74.51 71.58 75.88
 69.07 75.49 62.27 78.20 69.80 77.91 64.29 69.51 69.46 79.14
 72.73 74.09 70.94 68.84 74.59 68.91 77.15 65.73 75.99 71.36
 78.26 71.30 70.45 69.00 67.95 67.61 72.14 50.33 74.66 75.02
 76.11 69.48 73.88 77.91 74.55 73.05 69.32 76.50 75.04 69.77
 74.24 68.50 70.81 62.32 70.65 71.97 69.97 76.05 64.57 75.99
 73.77 64.71 68.30 58.44 76.86 62.24 72.66 78.74 78.54 70.51
 72.20 69.64 64.96 78.04 75.54 69.88 72.46 67.78 75.51 69.06
 75.74 71.86 75.50 75.28 79.14 78.32 78.64 69.95 74.91 69.89
 67.09 76.54 78.43 54.69 78.26 72.30 76.47 67.05 72.09 72.04
 70.43 72.20 77.28 73.86 70.63 73.04 74.54 69.90 75.50 68.89
 75.86 75.11 71.87 67.27 58.73 70.97 74.23 62.58 71.42 72.29
 69.92 78.93 69.20 68.89 76.47 77.82 72.63 70.40 73.76 69.94
 75.74 69.01 77.33 68.90 70.39 74.19 74.61 46.99 50.39 80.45

DATA 53

ORDER

1	1 OF 15.25	1 OF 17.07	12 OF 18.30
2	8 OF 20.20	18 OF 18.00	11 OF 15.50
3	6 OF 10.37	2 OF 10.22	
4	12 OF 16.47	1 OF 8.23	
5	15 OF 13.80		
6	5 OF 11.00	6 OF 14.00	6 OF 15.00
7	3 OF 15.30	5 OF 9.80	4 OF 11.00
	3 OF 14.10		4 OF 12.20
8	1 OF 10.90	3 OF 10.38	1 OF 15.11
	2 OF 17.00	2 OF 17.10	2 OF 17.20
	2 OF 17.40		2 OF 17.30
9	17 OF 12.19		
10	4 OF 10.97	4 OF 12.19	3 OF 14.02
11	1 OF 15.00	2 OF 14.00	6 OF 15.24
12	2 OF 14.70	1 OF 14.00	2 OF 12.00
	1 OF 12.20	1 OF 11.00	2 OF 11.00
13	7 OF 9.15	7 OF 11.00	1 OF 13.40
	6 OF 15.00		1 OF 12.80
14	10 OF 7.20		
15	30 OF 18.28		
16	5 OF 18.28		
17	2 OF 8.28	1 OF 7.88	1 OF 7.88
18	1 OF 7.47	3 OF 7.46	3 OF 8.26
	1 OF 7.37	1 OF 8.30	1 OF 8.44
	1 OF 8.24		5 OF 8.27
			1 OF 8.25

FIRST SET OF PARENT LENGTHS

67.23 81.68 85.96 68.59 84.95 59.32 87.36 91.31 92.61 64.80
 85.32 91.69 91.23 93.16 90.88 90.87 85.27 93.80 87.32 84.68
 61.20 81.07 67.62 88.08 83.90 84.69 68.99 92.70 67.28 93.32
 90.37 88.34 93.67 88.99 84.89 61.41 70.40 89.67 81.43 92.53
 87.25 91.72 84.07 85.94 92.97 91.40 81.61 85.05 88.80 88.38
 85.06 70.92 88.93 90.26 77.18 92.78 68.01 91.00 83.93 84.51
 86.68 93.52 92.97 84.12 77.04 84.09 92.20 89.33 92.15 91.94

SECOND SET OF PARENT LENGTHS

85.23	93.46	81.70	84.00	87.68	67.03	89.29	85.02	85.48	62.72
85.98	92.33	92.17	87.03	90.66	91.62	78.80	82.19	93.87	91.51
84.21	89.66	93.04	83.27	93.23	92.98	85.73	92.60	93.98	85.45
88.91	85.38	83.06	90.95	94.17	91.40	80.87	83.65	67.99	61.13
85.82	93.74	67.02	67.46	87.57	82.24	92.27	87.46	91.31	94.70
80.40	84.17	82.98	85.97	85.96	91.04	91.56	91.95	89.65	88.36
91.27	93.33	92.55	68.31	90.86	88.36	91.39	91.06	85.35	91.57

DATA 54

ORDER

1	3 OF 9.20	3 OF 11.00	3 OF 12.20	3 OF 14.10
	3 OF 15.30			
2	10 OF 21.00	30 OF 18.30	23 OF 15.30	
3	4 OF 9.80	2 OF 17.40	3 OF 12.20	3 OF 14.10
	3 OF 15.30			
4	4 OF 9.00	3 OF 12.00	3 OF 15.00	
5	2 OF 15.00	2 OF 14.00	2 OF 12.00	2 OF 11.00
6	4 OF 12.00	10 OF 14.00		
7	2 OF 12.15			
8	1 OF 12.00			
9	9 OF 18.28			
10	11 OF 15.24			
11	14 OF 12.19			
12	1 OF 9.29			
13	1 OF 7.81			

FIRST SET OF PARENT LENGTHS

79.78	70.40	69.36	80.52	83.84	80.32	79.36	76.28	83.35	86.61
86.49	71.39	83.94	85.12	84.27	87.50	80.51	84.60	84.94	71.25
86.03	82.91	80.13	76.58	81.26	82.34	79.66	86.91	84.49	82.78
81.25	82.44	84.33	70.06	68.32	81.39	83.43	78.09	84.68	77.23

SECOND SET OF PARENT LENGTHS

87.99	85.66	79.06	81.39	84.73	80.88	85.72	69.35	79.62	80.55
85.71	59.34	79.59	86.46	84.46	83.56	83.26	76.49	82.53	76.79
87.95	86.55	83.72	79.36	76.65	83.64	83.99	80.15	83.04	84.80
87.48	76.64	75.49	80.16	71.54	84.83	86.06	84.88	79.70	80.31

DATA 55

ORDER

1	4 OF 15.74	4 OF 13.05	1 OF 11.83	2 OF 6.68
2	1 OF 10.10			
3	2 OF 7.50	1 OF 7.05	1 OF 17.70	1 OF 10.07
	1 OF 13.25	2 OF 12.45		
4	17 OF 15.25			
5	2 OF 2.14			
6	2 OF 8.50	2 OF 8.66	2 OF 9.06	2 OF 9.22
	2 OF 8.76			
7	2 OF 16.30	1 OF 15.70		
8	9 OF 9.50			
9	12 OF 12.50	12 OF 7.65	1 OF 11.10	
10	1 OF 11.05	1 OF 11.05		
11	1 OF 18.28			
12	1 OF 15.77			
13	12 OF 12.25			
14	1 OF 10.64			
15	1 OF 13.55	3 OF 13.42	1 OF 8.82	1 OF 8.15
	2 OF 8.05	3 OF 8.00	2 OF 6.10	2 OF 6.00
	1 OF 10.86	1 OF 10.94	1 OF 10.11	1 OF 9.86
	1 OF 9.51	1 OF 9.21	1 OF 11.19	2 OF 10.71
16	2 OF 12.05			
17	1 OF 8.85			

18	1 OF	8.38							
19	1 OF	8.30							
20	2 OF	13.95	2 OF	14.55	1 OF	14.70			
21	1 OF	15.24	1 OF	18.28	1 OF	9.14	1 OF	18.28	
	2 OF	13.71	1 OF	12.19	1 OF	10.66	1 OF	18.28	
	1 OF	13.71							
22	1 OF	13.72	11 OF	12.19					
23	4 OF	13.72	14 OF	12.20	14 OF	9.15	2 OF	7.62	
24	2 OF	16.76	3 OF	18.28	1 OF	10.66	1 OF	12.19	
	1 OF	13.71	2 OF	15.24					
25	20 OF	18.28							
26	8 OF	17.06							
27	4 OF	18.28	4 OF	15.24	4 OF	12.19			
28	1 OF	13.67							
29	1 OF	13.71							
30	5 OF	7.01	1 OF	10.71					

FIRST SET OF PARENT LENGTHS

71.94	59.71	63.76	62.16	69.13	63.26	69.41	70.12	69.98	60.84
69.14	62.30	63.54	60.51	52.16	74.35	69.70	63.97	71.55	63.53
62.22	67.55	61.28	70.71	68.77	61.18	71.17	72.41	62.85	63.44
67.95	69.75	75.88	63.52	65.20	63.50	67.12	69.25	67.38	63.03
70.73	64.06	62.83	72.36	62.72	60.04	54.75	81.12	68.99	65.86
63.23	72.56	69.33	65.87	67.23	65.58	68.37	62.99	60.59	64.67

SECOND SET OF PARENT LENGTHS

62.47	52.48	59.58	63.17	64.78	69.14	68.82	68.65	56.46	64.94
65.48	64.37	68.89	52.98	66.85	63.24	63.73	67.25	68.90	64.14
65.69	65.87	60.02	71.25	64.27	68.38	52.77	60.25	68.54	70.91
61.64	82.60	68.83	65.54	68.78	57.76	64.55	61.45	71.76	63.46
63.89	65.23	68.34	61.94	68.60	70.09	56.18	68.23	57.70	65.16
66.64	69.34	69.29	60.69	71.46	63.58	69.25	68.32	63.60	59.75

DATA 56

ORDER

1	1 OF	14.50	2 OF	10.32	3 OF	10.71	4 OF	14.84	
2	1 OF	12.19	2 OF	15.24	2 OF	18.28			
3	2 OF	9.10							
4	4 OF	18.28	4 OF	15.24					

FIRST SET OF PARENT LENGTHS

53.60	46.83	55.59	46.05	58.00	53.54	46.21	67.41	46.59	55.81
-------	-------	-------	-------	-------	-------	-------	-------	-------	-------

SECOND SET OF PARENT LENGTHS

58.37	54.92	67.16	46.65	58.46	66.95	52.49	55.40	47.98	57.24
-------	-------	-------	-------	-------	-------	-------	-------	-------	-------

DATA 57

ORDER

1	3 OF	5.28	3 OF	15.49				
2	1 OF	12.82						
3	4 OF	11.58	4 OF	11.28	4 OF	11.28		
4	1 OF	12.49						
5	1 OF	10.80						
6	1 OF	16.20	1 OF	14.20	1 OF	11.52	1 OF	13.52
7	1 OF	13.40						
8	2 OF	18.75						
9	7 OF	9.95						
10	1 OF	11.41	1 OF	11.35	1 OF	11.52	1 OF	12.13
	1 OF	10.19	1 OF	10.16	1 OF	11.64	1 OF	11.57
	1 OF	10.73	1 OF	10.40	1 OF	10.29	1 OF	10.26
11	3 OF	12.20						
12	10 OF	19.30						
13	4 OF	12.20						
14	5 OF	18.99						
15	1 OF	13.42						
16	4 OF	19.00	2 OF	13.00				
17	1 OF	11.80						
18	1 OF	5.00						
19	4 OF	8.33						
20	16 OF	13.72	2 OF	7.62	4 OF	6.10	4 OF	18.30
21	4 OF	12.20	2 OF	6.10	1 OF	18.28	3 OF	18.28
	3 OF	18.28	2 OF	13.71	1 OF	12.19		
22	1 OF	18.28						
23	4 OF	18.28	4 OF	15.24				

FIRST SET OF PARENT LENGTHS

56.90	55.36	53.06	72.09	67.00	46.18	60.01	48.45	48.59	74.44
47.31	66.83	48.23	73.85	62.02	47.47	72.63	48.55	70.63	70.82
47.17	52.69	47.55	47.06	51.40	49.86	73.56	67.59	71.49	71.63
61.37	60.52	74.79	72.04	66.09	62.19	67.31	74.09	54.75	70.66

SECOND SET OF PARENT LENGTHS

57.13	47.12	61.52	70.03	68.51	70.75	53.92	68.73	43.11	61.06
68.36	67.55	62.06	72.40	55.87	52.59	54.68	71.77	47.46	51.87
46.03	70.32	60.61	46.82	47.38	49.87	60.76	49.77	53.72	56.39
49.82	48.45	46.26	67.50	48.67	67.50	53.92	72.00	71.67	70.98

DATA 58

ORDER

1	2 OF	7.80		
2	1 OF	10.08		
3	1 OF	17.70		
4	8 OF	5.80	3 OF	6.48
5	4 OF	11.82	4 OF	10.78
6	2 OF	13.65	2 OF	13.57
7	4 OF	7.97	10 OF	11.62
8	2 OF	8.90		
9	1 OF	22.55	1 OF	22.55
10	2 OF	7.02	4 OF	6.10
11	2 OF	12.30		
12	1 OF	13.42	1 OF	8.70

13	4 OF	19.90	2 OF	19.00	4 OF	15.95	4 OF	11.50
14	1 OF	18.45						
15	2 OF	11.72	2 OF	12.05	1 OF	7.57		
16	8 OF	10.73	8 OF	8.33				
17	2 OF	8.90						
18	4 OF	10.67						
19	1 OF	22.86	1 OF	7.62	1 OF	9.14		
20	1 OF	12.19	2 OF	15.24	2 OF	18.28		
21	2 OF	8.95	1 OF	9.04				
22	10 OF	9.10						
23	1 OF	13.06	1 OF	9.09	2 OF	11.59		

FIRST SET OF PARENT LENGTHS

65.75	42.16	47.20	43.77	66.77	65.73	44.42	61.94	63.80	44.32
64.67	46.19	65.03	60.52	52.79	59.04	65.09	42.18	48.28	44.00
45.48	62.86	60.80	65.05	62.13	64.31	45.66	45.13	64.87	45.03

SECOND SET OF PARENT LENGTHS

43.24	66.83	62.79	65.23	42.22	31.26	64.57	56.05	46.19	64.68
44.32	63.81	31.96	60.47	62.20	61.95	43.90	66.81	61.71	64.99
63.51	46.14	53.20	43.94	47.86	44.68	63.34	63.86	44.12	63.96

4.2 Programs

Program 12:- Final program to solve the Sequential Bar Cropping Problem by a Pattern Enumeration Method.

```

MASTER
COMMON MD,M,NS,NJ,NA,NIT(45),MS(6),MA(45,22),LA(45,22)
,IL(50),
IID(50),IP(50),SL,ST,SS,ITI
DIMENSION MAXT(7)
DATA MAXT/0,0,5,10,15,20,25/
C INPUT
READ (1,101)MD
DO 1 I=1,MD
READ (1,101)NO(I)
DO 1 J=1,NO(I)
READ (1,101) MA(I,J),X
1 LA(I,J)=X*100.0+1E-3
CALL SORT
DO 3 I=1,MD
3 WRITE(3,104)I,(MA(I,J),LA(I,J),J=1,NO(I))
DO 2 J=1,NO(I)
2 ID(J)=MA(I,J)
IL(J)=LA(I,J)
MS(1)=1
M=NO(1)
NO(1)=-NO(1)
NS=1
NJ=1
ST=0.0
SS=0.0
SL=0.0
ITI=0
ITABS=400
    
```

C ITERATION

```

9 READ(2,102)WW
  IWW=WW*100.0+1E-3
  IF(ID(1).GT.(MO-NU)*4+M*2)GOTO 10
  IT=IWW
  CALL PATTERN(IWW,IT,MAXT(NS+1)/2)
8 IF(NU.EQ.MO.OR.NS.GE.6)GOTO 4
  IF(IT.LE.10+MAXT(NS+2))GOTO 4
  IT=IWW
  NS=NS+1
  NU=NU+1
  DO 5 I=1,MO
    IF(NO(I).LE.0)GOTO 5
    DO 6 J=1,NO(I)
      ID(M+J)=MA(I,J)
      IL(M+J)=LA(I,J)
6 M=M+NO(I)
      MS(NS)=I
      NO(I)=-NO(I)
      CALL PATTERN(IWW,IT,MAXT(NS+1)/2)
      IF(IT.LE.MAXT(NS+1))GOTO 4
      NO(I)=-NO(I)
      M=M-NO(I)
5 CONTINUE
  DO 7 J=1,NO(NA)
    ID(M+J)=MA(NA,J)
7 IL(M+J)=LA(NA,J)
    M=M+NO(NA)
    MS(NS)=NA
    NO(NA)=-NO(NA)
    GOTO 8
10 CALL LOTS(IWW,IT)
    GOTO 11

```

C OUTPUT

```

4 IF(IT.GE.ITABS)CALL LEFTOVERS(IWW,IT)
11 CALL OUTPUT(IWW,IT)
  IF(NU.LT.MO.OR.M.GT.0)GOTO 9
  PT=1E2*ST/SL
  PS=1E2*SS/SL
  TC=PT+0.4*PS
  WRITE(3,103)ITI,SL,ST,PT,SS,PS,TC
  STOP
101 FORMAT(10,F0.0)
102 FORMAT(F0.0)
103 FORMAT(' NOS OF INGOTS=',I3/' TOTAL LENGTH OF INGOTS=',F
  1L SCRAP=',F6.2,' % SCRAP=',F5.2/' TOTAL TO STOCK=',F6.2,
  2,F5.2/' TOTAL % SCRAPPED=',F5.2)
104 FORMAT(I5,8(I5,' OF ',I5)/(5X,8(I5,' OF ',I5)))
  END
  SUBROUTINE SORT
  COMMON MO,M,NS,NU,NA,NO(45),MS(6),MA(45,22),LA(45,22),IL
  1IO(50),IP(50),SL,ST,SS,ITI
  COMMON /BLS/R(45)
  IF(MO.LE.1)RETURN
  DO 1 I=1,MO
    MM=NO(I)-1
8 K=0
    DO 5 J=1,MM
      IF(LA(I,J)-LA(I,J+1))6,7,5

```

9.2/' TOTA
% STOCK='

```

6      K=LA(I,J)
      LA(I,J)=LA(I,J+1)
      LA(I,J+1)=K
      K=MA(I,J)
      MA(I,J)=MA(I,J+1)
      MA(I,J+1)=K
      K=J
      GOTO 5
7      IF(LA(I,J).EQ.0)GOTO 5
      LA(I,J+1)=0
      MA(I,J)=MA(I,J)+MA(I,J+1)
      NO(I)=NO(I)-1
5      CONTINUE
      IF(K.GT.0)GOTO 8
      R(I)=0.0
      DO 9 J=1,NO(I)
9      R(I)=R(I)+MA(I,J)
1      R(I)=R(I)/(NO(I)*(1+LA(I,1)-LA(I,NO(I))))
      MM=MM-1
4      K=0
      DO 2 I=1,MM
      IF(R(I).GT.R(I+1))GOTO 2
      DO 3 J=1,MAX0(NO(I),NO(I+1))
      KN=MA(I,J)
      KL=LA(I,J)
      MA(I,J)=MA(I+1,J)
      LA(I,J)=LA(I+1,J)
      MA(I+1,J)=KN
3      LA(I+1,J)=KL
      A=R(I)
      R(I)=R(I+1)
      R(I+1)=A
      K=NO(I)
      NO(I)=NO(I+1)
      NO(I+1)=K
      K=I
2      CONTINUE
      MM=K-1
      IF(K.GT.1)GOTO 4
      RETURN
      END
      SUBROUTINE PATTERN(IWW,IT,ITMAX)
      COMMON MO,M,NS,NU,NA,NO(45),MS(6),MA(45,22),LA(45,22),IL
      ID(50),IP(50),SL,ST,SS,ITI
      COMMON /BLS/LSIZE(50),IS(50),ICF(50)
      IF(M.LE.0)RETURN
      MI=M+1
      LSIZE(M)=IWW
      ICF(1)=1
      IF(M.LE.1)GOTO 7
      DO 6 I=2,M
      J=MI-I
6      LSIZE(J)=MIN0(LSIZE(J+1),IL(J+1))
      KB=IL(M)
      DO 8 I=1,M
      IS(I)=0
      J=MI-I
      K=KB
      KB=IL(J)

```



```
9   KR=KB-K*(KB/K)
    KB=K
    K=KR
    IF(K.NE.0)GOTO 9
8   ICF(J)=KB
7   IW=IWW
    JP=0
1   JP=JP+1
    K=ICF(JP)
    IF(IW-K*(IW/K).GE.IT)GOTO 2
    IS(JP)=MIN0(IW/IL(JP),ID(JP))
    IW=IW-IS(JP)*IL(JP)
    IF(JP.LT.M.AND.IW.GE.LSIZE(JP))GOTO1
    IF(IW.GE.IT)GOTO2
    IT=IW
    NA=MS(NS)
    DO 5 I=1,M
5   IP(I)=IS(I)
    IF(IT.LE.ITMAX)RETURN
2   IF(IS(JP).GT.0)GOTO3
    JP=JP-1
    IF(JP.GT.0)GOTO2
    RETURN
3   IF(JP.EQ.M)GOTO4
    IS(JP)=IS(JP)-1
    IW=IW+IL(JP)
    GOTO1
4   IW=IW+IS(M)*IL(M)
    IS(M)=0
    GOTO2
    END
    SUBROUTINE LEFTOVERS(IW,IT)
    COMMON MD,M,NS,NJ,NA,NJ(45),MS(6),MA(45,22),LA(45,22),IL
1   ID(50),IP(50),SL,ST,SS,ITI                                (50),
    COMMON /BLS/ITW(6),MM(6)
    N=0
    IT=0
    DO 1 I=1,NS
    ITW(I)=0
    MM(I)=N
    DO 4 J=1,-NJ(MS(I))
    N=N+1
4   ITW(I)=ITW(I)+IL(N)*ID(N)
1   IT=IT+ITW(I)
    IM=900
    IF(IT.LE.IW.AND.NU.EQ.MD)IM=0
    IT=IW-IM
    DO 3 II=1,NS
    K=1
    DO 2 I=1,NS
2   IF(ITW(I).LT.ITW(K))K=I
    ITW(K)=1000000
    DO 3 J=MM(K)+1,MM(K)-NJ(MS(K))
    IP(J)=MIN0(ID(J),IT/IL(J))
3   IT=IT-IP(J)*IL(J)
    IT=IT+IM
    RETURN
    END
```

```

SUBROUTINE LOTS(IW,IT)
COMMON MD,M,NS,NU,NA,NO(45),MS(6),MA(45,22),LA(45,22),IL
(50),IP(50),SL,ST,SS,ITI
WRITE(3,201)
201 FORMAT(' PACKING PATTERN')
IW=IWW
NSP=NS
K=ID(1)
ID(1)=0
IDI=MIN0(IW/IL(1),K)
IW=IW-IDI*IL(1)
8 IT=IW+1
CALL PATTERN(IW,IT,5)
1 IF(IT.LE.50)GOTO 2
IF(NU.EQ.MD.OR.NS.GE.6)GOTO 3
IT=IW+1
NS=NS+1
NU=NU+1
DO 5 I=1,MD
IF(NO(I).LE.0)GOTO 5
DO 6 J=1,NO(I)
ID(M+J)=MA(I,J)
6 IL(M+J)=LA(I,J)

M=M+NO(I)
MS(NS)=I
NO(I)=-NO(I)
CALL PATTERN(IW,IT,5)
IF(IT.LE.10)GOTO 2
NO(I)=-NO(I)
M=M-NO(I)
5 CONTINUE
IF(IT.GE.IW)GOTO 10
DO 7 J=1,NO(NA)
ID(M+J)=MA(NA,J)
7 IL(M+J)=LA(NA,J)
M=M+NO(NA)
MS(NS)=NA
NO(NA)=-NO(NA)
GOTO 1
3 IF(NS.LE.NSP)GOTO 4
NO(MS(NS))=-NO(MS(NS))
M=M-NO(MS(NS))
10 NS=NS-1
NU=NU-1
GOTO 3
4 IDI=IDI-1
IW=IW+IL(1)
IF(IDI.GT.0)GOTO 8
ID(1)=K
IT=1000000
CALL PATTERN(IWW,IT,100)
IF(IT.LE.400)RETURN
IT=IWW-900
DO 9 I=1,M
IP(I)=MIN0(IT/IL(I),ID(I))
9 IT=IT-IP(I)*IL(I)
IT=IT+900
RETURN
```

```

2      IP(1)=ID1
      ID(1)=K
      RETURN
      END
      SUBROUTINE OUTPUT(IW,IT)
      COMMON MD,M,NS,NJ,NA,NQ(45),MS(6),MA(45,22),LA(45,22),IL
1      ID(50),IP(50),SL,ST,SS,ITI          (50),
      WRITE(3,101)IW,IT,M,(MS(I),I=1,NS)
      IF(IT.LT.900)ST=ST+IT*1E-2
      IF(IT.GE.900)SS=SS+IT*1E-2
      SL=SL+IW*1E-2
      ITI=ITI+1
      NR=0
      J=1
      IC=-NQ(MS(1))
      I=0
1      I=I+1
      IF(I.GT.M+NR)RETURN
      IF(I.LE.IC)GOTO 3
      J=J+1
      IC=IC-NQ(MS(J))
3      ID(I-NR)=ID(I)-IP(I)
      IL(I-NR)=IL(I)
      IF(ID(I-NR).GT.0)GOTO 1
      M=M-1
      NR=NR+1
      NQ(MS(J))=NQ(MS(J))+1
      IF(NQ(MS(J)).LT.0)GOTO 1
      NS=NS-1
      IF(J.GT.NS)RETURN
      DO 2 K=J,NS
2      MS(K)=MS(K+1)
      IC=IC-NQ(MS(J))
      GOTO 1
101  FORMAT(' LENGTH=',I5,' TRIM=',I4,' M=',I2,' ORDERS=',I3)
      END
      FINISH

```

- Input data:-
1. The number of orders (integer) from input 1.
 2. A list of the orders; each order containing the following information:-
 - (a) The number of lengths in the order (integer) from input 1.
 - (b) A list of the quantities (integer) and lengths (real) required from input 1.
 3. A list of parent lengths (real) from input 2.

4.3 Results

Table 33 Summary of the schedules obtained by manual methods to the initial five sets of data.

Data	Number of girders used	% trim	% off-cut	Total % scrapped
39	45	.71	1.83	1.44
40	22	1.52	3.68	2.99
41	70	.48	.84	.82
42	15	.51	.32	.64
43	45	.84	2.21	1.72
Overall*	197	.71	1.59	1.35

* These overall figures are not simple averages as the number of parent girders, average length of a parent girder and weight/m vary between data sets.

Table 34 Summary of the schedules obtained by the British Steel heuristic to the initial five sets of data.

Data	Number of girders used	% trim	% off-cut	Total % scrapped
39	45	.46	2.09	1.30
40	22	1.22	3.99	2.82
41	70	.46	.87	.81
42	16	.37	7.67	3.44
43	44	.69	.19	.77
Overall	197	.58	1.66	1.24

The results shown in the tables above are taken from Table 2, p. 16 in (8). The method here called the British Steel Heuristic being that using the best overall parameter set (referred to as 1(B) in (8)). This method has been used for comparison purposes as 1(A) requires manual reallocation of stock lengths and 1(C) uses the best parameters for each set of data which, in practice, would not be known.

Table 35 Summary of the schedules obtained by the pattern enumeration method without sorting the orders.

Data	Number of girders used	% trim	% off-cut	Total % scrapped
39	45	0.54	2.01	1.34
40	22	1.04	4.18	2.71
41	71	0.18	2.64	1.24
42	15	0.06	0.78	0.37
43	45	0.22	2.83	1.35
Overall	198	0.31	2.60	1.35

Table 36 Summary of the schedules obtained by the pattern enumeration method when the orders are sorted by the total number of pieces required.

Data	Number of girders used	% trim	% off-cut	Total % scrapped
39	45	0.26	2.29	1.17
40	22	1.04	4.18	2.71
41	70	0.09	1.23	0.58
42	15	0.06	0.77	0.37
43	44	0.14	0.74	0.43
Overall	196	0.21	1.45	0.78

Table 37 Summary of the schedules obtained by the pattern enumeration method when the orders are sorted by Evans & Quarrington's criteria.

Data	Number of girders used	% trim	% off-cut	Total % scrapped
39	45	0.25	2.30	1.17
40	22	1.04	4.18	2.71
41	70	0.05	1.27	0.56
42	15	0.05	0.79	0.36
43	44	0.19	0.69	0.47
Overall	196	0.20	1.45	0.78

Table 38 Summary of the schedules obtained by the British Steel Heuristic to the full twenty sets of data.

Data	Number of girders used	% trim	% off-cut	Total % scrapped
39	45	0.67	1.88	1.42
40	22	1.22	3.99	2.82
41	70	0.57	0.75	0.87
42	16	0.37	7.67	3.44
43	45	0.69	2.35	1.63
44	260	0.45	0.58	0.68
45	180	0.65	1.24	1.15
46	69	0.50	3.47	1.89
47	100	0.39	0.41	0.55
48	67	1.10	3.59	2.54
49	11	0.42	7.79	3.54
50	69	0.34	0.47	0.53
51	158	0.59	0.73	0.88
52	147	0.72	0.32	0.85
53	49	0.51	1.35	1.05
54	31	0.96	1.19	1.44
55	48	0.99	0.90	1.35
56	8	1.11	15.06	7.13
57	31	0.92	1.10	1.36
58	25	0.78	2.37	1.73
Overall	1451	0.65	1.48	1.24

Table 39 Summary of the schedules obtained by the pattern enumeration method with a varying acceptable trim parameter.

Data	Number of girders used	% trim	% off-cut	Total % scrapped
39	46	0.26	4.50	2.06
40	22	1.04	4.18	2.71
41	70	0.05	1.27	0.56
42	15	0.05	0.79	0.36
43	44	0.19	0.69	0.47
44	258	0.06	0.18	0.13
45	178	0.26	0.43	0.43
46	67	0.18	0.72	0.47
47	100	0.04	0.77	0.34
48	70	0.89	8.12	4.13
49	11	0.27	7.94	3.45
50	69	0.04	0.77	0.35
51	157	0.06	0.57	0.29
52	146	0.08	0.28	0.19
53	49	0.10	1.75	0.80
54	31	0.23	1.91	1.00
55	48	0.20	1.69	0.87
56	8	0.99	15.18	7.06
57	31	0.69	1.32	1.22
58	25	0.14	3.01	1.34
Overall	1445	0.20	1.63	0.86

Table 40 Summary of the schedules obtained by the pattern enumeration method when orders for large quantities are forced into solution.

Data	Number of girders used	% trim	% off-cut	Total % scrapped
39	45	0.28	2.27	1.19
40	22	0.98	4.23	2.68
41	70	0.09	1.23	0.58
42	15	0.07	0.77	0.38
43	44	0.14	0.74	0.44
44	258	0.06	0.17	0.13
45	178	0.32	0.36	0.47
46	67	0.34	0.57	0.57
47	100	0.06	0.74	0.35
48	65	0.44	1.40	1.00
49	11	0.30	7.91	3.47
50	69	0.04	0.78	0.35
51	157	0.09	0.55	0.31
52	146	0.11	0.25	0.21
53	49	0.10	1.75	0.80
54	31	0.36	1.78	1.08
55	48	0.20	1.69	0.87
56	8	0.99	15.18	7.06
57	31	0.53	1.49	1.13
58	25	0.25	2.89	1.41
Overall	1439	0.20	1.14	0.65

Table 41 Summary of the schedules obtained by the pattern enumeration method with a modified rule for forcing large orders into solution.

Data	Number of girders used	% trim	% off-cut	Total % scrapped
39	45	0.18	2.37	1.13
40	22	0.98	4.23	2.68
41	70	0.09	1.23	0.58
42	15	0.07	0.77	0.38
43	44	0.12	0.76	0.43
44	258	0.06	0.17	0.13
45	178	0.30	0.39	0.45
46	68	0.24	2.23	1.13
47	100	0.06	0.74	0.36
48	65	0.44	1.40	1.00
49	11	0.30	7.91	3.47
50	69	0.04	0.77	0.35
51	157	0.07	0.56	0.30
52	146	0.12	0.24	0.22
53	49	0.10	1.75	0.80
54	31	0.23	1.91	1.00
55	48	0.20	1.69	0.87
56	8	0.99	15.18	7.06
57	31	0.37	1.65	1.03
58	25	0.25	2.89	1.41
Overall	1440	0.18	1.21	0.66

Table 42 Summary of the schedules obtained by the modified British Steel Heuristic.

Data	Number of girders used	% trim	% off-cut	Total % scrapped
39	45	0.32	2.23	1.21
40	22	0.52	4.70	2.40
41	70	0.15	1.17	0.62
42	15	0.13	0.71	0.41
43	44	0.25	0.63	0.50
44	259	0.23	0.38	0.38
45	179	0.31	0.93	0.68
46	69	0.36	3.60	1.80
47	100	0.12	0.68	0.39
48	65	1.05	0.79	1.37
49	11	0.30	7.91	3.46
50	69	0.11	0.70	0.39
51	157	0.25	0.39	0.41
52	146	0.21	0.15	0.27
53	49	0.22	1.64	0.88
54	31	0.36	1.78	1.07
55	48	0.24	1.65	0.90
56	8	1.70	14.47	7.50
57	32	0.29	4.86	2.23
58	25	0.24	2.90	1.40
Overall	1444	.30	1.36	.84

Table 43 Summary of the schedules obtained by the British Steel Heuristic using the second set of parent girders.

Data	Number of girders used	% trim	% off-cut	Total % scrapped
39	45	0.94	1.08	1.37
40	25	1.55	5.73	3.84
41	69	0.58	1.17	1.05
42	16	0.42	5.43	2.59
43	45	1.15	2.75	2.25
44	261	0.45	0.47	0.64
45	176	0.47	2.28	1.38
46	69	0.82	1.57	1.45
47	100	0.30	0.52	0.51
48	70	0.90	6.65	3.56
49	11	0.43	5.99	2.83
50	70	0.40	0.45	0.58
51	156	0.41	0.37	0.56
52	149	0.66	0.64	0.92
53	48	0.52	1.23	1.01
54	31	0.69	2.08	1.52
55	49	0.96	0.84	1.30
56	7	0.32	11.25	4.82
57	32	0.86	2.20	1.74
58	27	0.46	7.82	3.59
Overall	1456	0.61	1.90	1.37

Table 44 Summary of the schedules obtained by the pattern enumeration method using the second set of parent girders.

Data	Number of girders used	% trim	% off-cut	Total % scrapped
39	45	0.50	1.52	1.11
40	26	1.33	10.44	5.51
41	68	0.09	0.30	0.21
42	16	0.14	5.71	2.42
43	44	0.14	1.45	0.72
44	259	0.06	0.14	0.12
45	173	0.08	1.00	0.48
46	68	0.34	0.61	0.58
47	100	0.05	0.76	0.36
48	66	0.76	1.22	1.25
49	11	0.37	6.06	2.79
50	70	0.05	0.81	0.37
51	156	0.11	0.67	0.38
52	148	0.12	0.49	0.31
53	48	0.12	1.63	0.77
54	31	0.11	2.66	1.18
55	49	0.13	1.67	0.80
56	7	1.62	9.95	5.60
57	31	0.45	0.00	0.45
58	25	0.26	0.74	0.56
Overall	1441	0.20	1.15	0.66

Table 45 Comparison of the total amount of scrap (%) by different scheduling methods.

Method Data	A	B	C	D	E	F	G	H	I	J	K	L
39	1.44	1.30	1.34	1.17	1.17	1.42	2.06	1.19	1.13	1.21	1.37	1.11
40	2.99	2.82	2.71	2.71	2.71	2.82	2.71	2.68	2.68	2.40	3.84	5.51
41	0.82	0.81	1.24	0.58	0.56	0.87	0.56	0.58	0.58	0.62	1.05	0.21
42	0.64	3.44	0.37	0.37	0.36	3.44	0.36	0.38	0.38	0.41	2.59	2.42
43	1.72	0.77	1.35	0.43	0.47	1.63	0.47	0.44	0.43	0.50	2.25	0.72
44						0.68	0.13	0.13	0.13	0.38	0.64	0.12
45						1.15	0.43	0.47	0.45	0.68	1.38	0.48
46						1.89	0.47	0.57	1.13	1.80	1.45	0.58
47						0.55	0.34	0.35	0.36	0.39	0.51	0.36
48						2.54	4.13	1.00	1.00	1.37	3.56	1.25
49						3.54	3.45	3.47	3.47	3.46	2.83	2.79
50						0.53	0.35	0.35	0.35	0.39	0.58	0.37
51						0.88	0.29	0.31	0.30	0.41	0.56	0.38
52						0.85	0.19	0.21	0.22	0.27	0.92	0.31
53						1.05	0.80	0.80	0.80	0.88	1.01	0.77
54						1.44	1.00	1.08	1.00	1.07	1.52	1.18
55						1.35	0.87	0.87	0.87	0.90	1.30	0.80
56						7.13	7.06	7.06	7.06	7.50	4.82	5.60
57						1.36	1.22	1.13	1.03	2.23	1.74	0.45
58						1.73	1.34	1.41	1.41	1.40	3.59	0.56
Initial five sets of data.	1.35	1.24	1.35	0.78	0.78	1.54	0.92	0.79	0.78	0.80	1.81	1.13
Overall						1.24	0.86	0.65	0.66	0.84	1.37	0.66

Key to methods

- A - Manual scheduling.
 - B - Initial British Steel heuristic.
 - C - Pattern enumeration, without sorting orders.
 - D - " " orders sorted by the total number of pieces in an order.
 - E - " " " " " Evans & Quarrington criteria.
 - F - Intermediate British Steel heuristic.
 - G - Pattern enumeration with varying acceptable trim.
 - H - " " " routine to force high demands into solution.
 - I - " " " second set of acceptable trim parameters.
 - J - Final British Steel heuristic.
 - K - Intermediate British Steel heuristic.
 - L - Final pattern enumeration method.
- } second set of parent lengths.

APPENDIX 5. BIBLIOGRAPHY

The bibliography has been split into two sections. Section A contains works which are referenced explicitly in the text (i.e. either basic methods or background material to one of the case studies). Section B contains works which, while not of direct relevance to the development of the pattern enumeration technique, do show different methods for tackling the trim loss problem. General theory texts have also been included in this section.

SECTION A.

1. A. G. Begeed-Dov, 'Lower and Upper Bounds for the Number of Lattice Points in a Simplex', Society for Industrial and Applied Mathematics Journal 22(1) (1972) 106-8.
2. R. H. Bernhard, 'Use of Inventories for Reduction of Trim Losses', Journal of Industrial Engineering 18 (1967) 668-70.
3. F. A. Caruso & J. J. Kokat, 'Reducing Coil-Slitting Waste', Western Electric Engineer 17(1) (1973) 12-17.
4. I. Coverdale & F. Wharton, 'An Improved Heuristic Procedure for a Nonlinear Cutting Stock Problem', Management Science 23(1) (1976) 78-86.
5. I. Coverdale & F. Wharton, 'On Cutting Stock Problems', viewpoint and reply by R. S. Stainton in Journal of the Operational Research Society 29(5) (1978) 503-506.
6. K. Eisemann, 'The Trim Problem', Management Science 3(3) (1957) 279-84.
7. G. J. Evans, Results of Beam Mill Sawman Simulation and Preliminary Results of Cut-up Optimisation Routine, British Steel Corporation Teesside & Workington Group, Systems and Operational Research Department Report No. OR78/1 (1973).
8. G. J. Evans, Results of an Improved Cut-up Optimisation Routine for No. 10 Mill, British Steel Corporation Teesside & Workington Group, Systems and Operational Research Department Report No. OR78/2 (1974).
9. P. C. Gilmore & R. E. Gomory, 'A Linear Programming Approach to the Cutting-stock Problem', Operations Research 9(6) (1961) 849-59.

10. P. C. Gilmore & R. E. Gomory, 'A Linear Programming Approach to the Cutting Stock Problem - Part 2' Operations Research 11(6) (1963) 863-88.
11. R. W. Haessler, 'An Application of Heuristic Programming to a Non-linear Cutting Stock Problem occurring in the Paper Industry' (University of Michigan Ph.D., 1968).
12. R. W. Haessler, 'A Heuristic Programming Solution to a Non-linear Cutting Stock Problem', Management Science 17(12) (1971) B793-B802.
13. E. C. Johns, 'Heuristic Procedures for Solving the Paper Trim Problem', from J. F. Pierce (ed.), Operations Research and the Design of Management Information Systems, New York (1967) Technical Association of the Pulp and Paper Industry (of America) Special Technical Association Publication Series No. 4, ch. 20, 361-9.
14. L. V. Kantorovich, 'Mathematical Methods of Organizing and Planning Production', Management Science 6(4) (1960) 363-422.
15. R. Marconi, 'Heuristic Method for Minimizing Trim Loss in the Paper Industry', International Business Machines Technical Disclosure Bulletin 14(1) (1971) 325-7.
16. J. F. Pierce, Some Large Scale Production Scheduling Problems in the Paper Industry, Eaglewood Cliffs, New Jersey, (1964) Prentis Hall.
17. J. F. Pierce (ed.), Operations Research and the Design of Management Information Systems New York (1967) Technical Association of the Pulp and Paper Industry (of America), Special Technical Association Publication Series No. 4.
18. P. Sanders, 'Computer Control of the Finishing End of a Medium-section Finishing Mill', Journal of the West of Scotland Iron and Steel Institute 81 (1973).
19. R. S. Stainton, 'The Cutting Stock Problem for the Stockholder of Steel Reinforcement Bars', Operational Research Quarterly 28 (1ii) (1977) 139-149.
20. C. B. Tilanus & C. Gerhardt, 'An Application of Cutting Stock in the Steel Industry', from K. B. Haley (ed.) 'Operational Research 75', (1976) North Holland Publishing Company.

21. S. Vajda, Readings in Mathematical Programming London (1958) Pitman.

22. M. K. Wig, 'Diophantine Programming and the Stock Cutting Problem' (North State University, Raleigh Ph.D., 1970).

SECTION B.

23. R. B. Banerji, Theory of Problem Solving New York (1969) Elsevier.

24. S. Barnett & G. J. Kynch, 'Exact Solution of a Simple Cutting Problem', Operations Research 15(6) (1967) 1051-6.

25. C. R. Bector, 'Programming Problems with Convex Fractional Functions', Operations Research 16(2) (1968) 383-91.

26. A. V. Cabot, 'An Enumeration Algorithm for Knapsack Problems', Operations Research 18(2) (1970) 306-11.

27. A. Charnes & W. W. Cooper, 'Management Models and Industrial Applications of Linear Programming', Management Science 4(1) (1957) 38-91.

28. G. D. Cohen, 'Comments on a paper by M. L. Wolfson: "Selecting the Best Lengths to Stock" ', Operations Research 14(2) (1966) 341.

29. G. B. Dantzig, 'Discrete-Variable Extremum Problems', Operations Research 5(2) (1957) 266-77.

30. G. B. Dantzig & P. Wolfe, 'Decomposition Principle for Linear Programs', Operations Research 8(1) (1960) 101-11.

31. G. B. Dantzig, Linear Programming and Extensions Princeton New Jersey (1963) Princeton University Press.

32. J. M. Dutton, 'Production Scheduling - a Behavioural Model', International Journal of Production Research 3(1) (1964) 3-27.

33. R. G. Dyson, 'Cutting Stock Problems in the Glass Industry', Paper at the 8th Annual Research Conference, Lancaster (1973).

34. R. G. Dyson & A. S. Gregory, 'The Cutting Stock Problem in the Flat Glass Industry', Paper at the 8th Annual Research Conference Lancaster (1973); or in Operational Research Quarterly 25(1) (1974) 41-53.

35. N. V. Findler & B. Meltzer, Artificial Intelligence and Heuristic Programming Edinburgh (1971) Edinburgh University Press.

36. P. C. Gilmore, 'The Cutting Stock Problem', Canadian Mathematical Bulletin 9 (1966) 737-9.
37. P. C. Gilmore, 'An Algorithm helpful in the Solution of Integer One Dimensional Cutting Stock Problems', International Business Machines, Report RC3389 (1971).
38. P. C. Gilmore & R. E. Gomory, 'Multistage Cutting Stock Problems of Two and More Dimensions', Operations Research 13(1) (1965) 94-120
39. P. C. Gilmore & R. E. Gomory, 'The Theory and Computation of Knapsack Functions', Operations Research 14(6) (1966) 1045-74.
40. R. E. Gomory, 'The Trim Problem', International Business Machines Systems Journal 1 (1962) 77-82.
41. R. E. Gomory, 'Large and Nonconvex Problems in Linear Programming', Proceedings of Symposia in Applied Mathematics 15 American Mathematical Society (1963) 125-39.
42. R. E. Gomory, 'An Algorithm for Integer Solutions to Linear Programs', Princeton International Business Machines Mathematics Research Project Technical Report 1 (1958); or in R. L. Graves & P. Wolfe (eds.), Recent Advances in Mathematical Programming New York (1963) McGraw Hill 269-302.
43. S. G. Hahn, 'On the Optimal Cutting of Defective Sheets', Operations Research 16(6) (1968) 1100-14.
44. M. J. Haims & H. Freeman, 'A Multistage Solution of the Template-layout Problem', Institute of Electrical and Electronics Engineers Transactions on Systems Science and Cybernetics SSC6(2) (1970) 145-51.
45. R. R. Hodges. 'The Trim Problem Reconsidered', from J. F. Pierce (ed.), Operations Research and the Design of Management Information Systems New York (1967) Technical Association of the Pulp and Paper Industry (of America) Special Technical Association Publication Series No. 4, ch. 18, 355-9.
46. R. M. Karp & M. Held, 'Finite-state Processes and Dynamic Programming', Society for Industrial and Applied Mathematics Journal 15(3) (1967) 693-718.

47. H. P. Kunzi, H. G. Tzschach, & C. A. Zehnder, Numerical Methods of Mathematical Optimization New York (1968) Academic Press.
48. E. L. Lawler & D. E. Wood, 'Branch-and-Bound Methods: A Survey' Operations Research 14(4) (1966) 699-719.
49. G. B. Mathews, 'On the Partition of Numbers', Proceedings of the London Mathematical Society 28 (1897) 486-490.
50. H. W. van den Meerendonk & J. H. Schouten, 'Trim-losses in the Manufacture of Corrugated Cardboard', Philips Technical Review 24(4/5) (1963) 121-9.
51. N. J. Nilsson, Problem Solving Methods in Artificial Intelligence New York (1971) McGraw-Hill.
52. C. C. Pegels, 'A Comparison of Scheduling Models for Corrugator Production', Journal of Industrial Engineering 18 (1967) 466-72.
53. J. F. Pierce, 'A Bibliography and KWIC Index of Operations Research and Related Topics in the Pulp and Paper Industry'. Technical Association of the Pulp and Paper Industry (of America), Special Report No. 436 New York (1965).
54. J. F. Pierce, 'On the Solution of Integer Cutting Stock Problems by Combinatorial Programming-Part 1', International Business Machines Technical Report 36.Y02 Cambridge, Massachusetts (1966).
55. C. C. Poirier, 'Automated Data Processing for the Corrugated Box Plant' from J. F. Pierce (ed.) Operations Research and the Design of Management Information Systems New York (1967) Technical Association of the Pulp and Paper Industry (of America) Special Technical Association Publication Series No. 4, ch. 24, 416-32.
56. C. N. Potts, 'Production Scheduling and Stock Control at George L. Scott & Company Limited' (Hull M.Sc. in Operational Research, 1971).
57. J. F. Shapiro, 'Dynamic Programming Algorithms for the Integer Programming Problem-1: The integer programming problem viewed as a knapsack type problem', Operations Research 16(1) (1968) 103-21.
58. H. A. Simon & A. Newell, 'Heuristic Problem Solving: The Next Advance in Operations Research', Operations Research 6(1) (1958) 1-10.

59. F. M. Tonge, 'The Use of Heuristic Programming in Management Science', Management Science 7(3) (1961) 231-7.
60. S. Vajda, 'Trim Loss Reduction', in Planning by Mathematics London (1969) Pitman, ch. 10, 54-66.
61. H. M. Weingartner, 'Municipal Bond Coupon Schedules with Limitations on the Number of Coupons', Management Science 19(4) (Applications) (1972) 369-78.
62. M. L. Wolfson, 'Selecting the Best Lengths to Stock', Operations Research 13(4) (1965) 570-85.