Oxford University Computing Laboratory

Computer Manuals

ICL 1900 Scientific and Technical Software
Mathematical Programming Part 5
Trim Loss - One Dimensional

3343/5

SCIENTIFIC & TECHNICAL SOFTWARE

Mathematical Programming. Part 5

Trim Loss. One Dimensional

1900

ICL

International Computers Limited
©

User notice

MANUAL (NOTICE NO.) 5/2/69

3343/5 TRIM LOSS - ONE DIMENSIONAL (1)

File one copy of this notice with each of the manuals indicated.

#XUT1/4, #XUT2/5: TRIM LOSS ONE DIMENSIONAL

The output from these programs has been altered to conform with the example in the manual.

FORM 1/230/45(10.68)

Issued by Technical Publications Service, 30 - 31 Friar Street, Reading, Berkshire. RG1 1DX

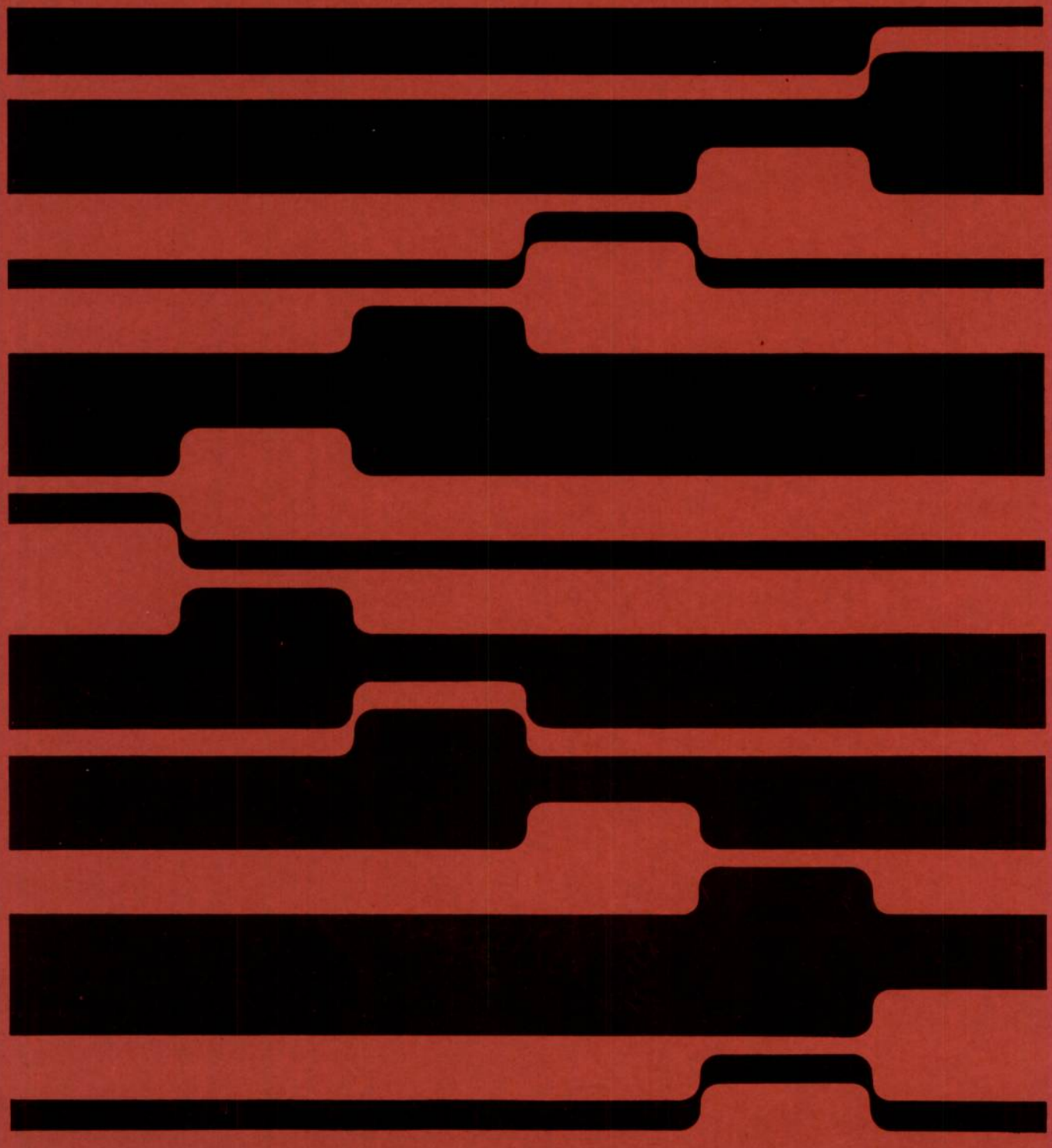
ICL

Scientific
and
technical
software

1900 Series

Mathematical programming

Part 5
trim loss
one dimensiona



ICL

Scientific
and
technical
software

1900 Series

Mathematical programming

Part 5
trim loss
one dimensional

With effect from 9th July 1968 the name of International Computers and Tabulators Limited has been changed to International Computers Limited.

Technical Publication 3343/5

First Edition June 1966

Reprinted May 1967
(incorporating Amendment List 1
with the exception of hand amendments)

Issued by Technical Publications Service
International Computers Limited
Head Office: ICL House, Putney, London SW15
and printed in Great Britain by
ICL Printing Services, Letchworth, Hertfordshire

1900 SERIES SCIENTIFIC AND TECHNICAL SOFTWARE

MATHEMATICAL PROGRAMMING

Part 5: Trim Loss - One Dimensional

First Edition June 1966

Amendment List No. 1

Each amendment list contains one or more numbered amendments. These amendments are provided in the form of instructions to replace one or more existing pages, to add one or more new pages, or to alter an existing page.

When all additions and alterations have been made, the contents list and amendment record should be updated, and the amendment list should be filed at the back of the manual.

The following conventions are adopted in amendment lists:

1 Line number 1 is the first line following the section heading or, if no section reference is given, the first line of the page.

2 Line number -1 is the last line of the section or, if no section reference is given, the last line of the page.

3 *All* printed lines, including headings or mathematical expressions, are taken into account when a line number is determined.

4 The date of issue appears at the foot of all extra pages and all re-issued pages in the form (month, year).

When a page is re-issued, the passages which have been materially altered contain a vertical mark in the outside margin.

5 When the recipient is asked to make a minor amendment to a page, he should note the amendment list number and the amendment number in the outside margin by writing, for example, AL3/2 to identify amendment 2 of amendment list 3.

Item	*Part*	*Chapter*	*Page*	*Section*	*Line*	*Amendment*
2	5			CONTENTS	-2	Insert after this line: '3.3 SPECIAL INSTRUCTION FOR A SHORT RUN - - - - 9'
3	5		1	INTRODUCTION	3	Amend the program names 'X2T1 and X2T2' to 'XUT1 and XUT2'.
4	5		1	INTRODUCTION	17	Amend the program name 'X2T2' to 'XUT2'.

International Computers and Tabulators Limited
I.C.T. House, Putney, London, S.W.15

Amendment List No. 1

Item	*Part*	*Chapter*	*Page*	*Section*	*Line*	*Amendment*
5	5		3	INTRODUCTION	-6	Amend the line to read: 'At present, both programs will accept up to 38 different order sizes and ...'
6	5		3	INTRODUCTION	-5	Amend the line to read: ' .... They require approximately 12K words of storage ....'
7	5	1	5	1.2	5	Amend the value in brackets for parameter m to read '<39'.
8	5	1	6	1.3	-6	Amend the line to read: 'on paper tape. Each stock length must be followed by a space and its cost punched as'

Part 5

TRIM LOSS ONE DIMENSIONAL

I·C·T 1900 series

CONTENTS

3343(5.66)

INTRODUCTION

The Trim Loss or Cutting Stock problem is to fill, at minimum cost, an order for various quantities of material of different sizes where these quantities are to be cut from pieces of stock of given size and given cost. This manual describes two optional programs, X2T1 and X2T2, each of which provides a solution to the one dimensional version of the problem, in which *lengths* of material are to be cut.

Each program must be provided with input data specifying the order sizes and quantities required, and the stock sizes available together with their cost per unit length. From this data, the program will calculate a series of cutting patterns, each more economical than the last. When improvement ceases or becomes insignificant, full details of the cutting pattern and costs are printed.

In general, it will be economical to use a certain stock size only if a significant number of order lengths are to be cut from pieces of that size. Each program can satisfy this requirement by excluding from the cutting pattern any stock size which falls below a minimum usage level specified by the user.

The programs can also take account of cutting procedures. Stock is often cut by a single movement of multiple knives rather than multiple cuts by a single knife. In these circumstances, the number of knives available obviously limits the number of pieces to be cut from a given stock length. Any such limitation to the number of knives can be input as data, and the program will then provide a cutting pattern which satisfies the limitation.

The second program, X2T2, *always* provides a cutting pattern in which at most two sizes of material will be cut from any given stock length at any given time. This is necessary in several organizations where a secondary operation is to be carried out on the cut pieces and only two machines are available, each adjusted to receive a certain size of material.

Trim Loss can be expressed as a conventional linear programming problem in which the variables are restricted to integers. However, the large number of variables involved generally makes computation infeasible by standard linear programming techniques, even if an approximate solution is being sought. While it is unnecessary for the user to understand the technique employed by the programs to overcome this difficulty, the following brief description is provided for the benefit of readers experienced in linear programming.

We assume that there are stock lengths L_1, L_2 .... L_k of uniform material available in unlimited quantities. These are to be cut into pieces to fill incoming orders. Each such order consists of a request for a given number of pieces N_i of length l_i ($i = 1, 2, \ldots m$). Provided there is some stock length $L_j \geq l_i$ for all i, it is possible to fill the order.

If costs are assigned to the stock lengths, and the cost of filling the order is taken as the total cost of all stock pieces cut, then the problem is to fill the order at least cost.

An 'activity' is defined as the cutting of a specified stock length in a specified manner. Thus, if there is a stock length of 11 units, a typical activity would be

3 lengths of 2 units

1 length of 4 units

1 length of trim or waste.

3343(5.66)

We assign variables $x_1, x_2, \ldots x_n$, each variable corresponding to the possible activity of cutting $l_1, l_2 \ldots l_m$ from $L_1, L_2 \ldots L_k$, the variable x_j being the number of lengths cut by activity j. These variables must satisfy the following inequalities:

$$a_{i1} x_1 + a_{i2} x_2 + \ldots + a_{in} x_n \geq N_i \quad (i = 1,2 \ldots m)$$

Thus, for each length l_i, the j^{th} activity is used x_j times and cuts a_{ij} pieces of l_i. This contributes $a_{ij} x_j$ lengths of l_i to the sum for l_i, and this sum must be $\geq N_i$ for all j if the order is to be filled.

Now, if c_j is the cost of the stock length used in the j^{th} activity, then the cost function to be minimized is:

$$c_1 x_1 + c_2 x_2 + \ldots + c_n x_n$$

It is necessary to transform the m inequalities into equations and this can be done in two ways. We can introduce m slack variables x_{n+i} $(i = 1 - m)$, or we can remove the restriction to integers of the variables $x_1, x_2 \ldots x_n$.

If we adopt the second approach, we will arrive at a solution where a particular cutting pattern must be repeated, say, 25.7 times. In practice, this means we must cut either 25 or 26 stock lengths in a given way. If we cut 25 lengths, we will not fill the order; if we cut 26, we will overfill the order. However, the increase in cost caused by rounding up will be small, particularly if the initial non-integral answer is large, and the program adopts this procedure of rounding up.

In fact, slack variables are also introduced, for without them every minimal solution will be in terms of exactly m activities when there might be a better solution with less than m activities. Although the introduction of a slack variable implies that it is cheaper to overfill a particular order length and regard the excess as waste, this waste may in practice be saleable.

Since there are so many activities satisfying the restrictions of the problem, the solution by standard linear programming techniques becomes impracticable. The program therefore applies the 'Decomposition Principle for Linear Programs'. When, in the Simplex method, we reach the stage of looking for an improving column (corresponding to an activity) instead of selecting from a large collection of columns the column which gives maximum improvement, we calculate a useful column by solving an auxiliary problem.

Consider the matrix A pre-multiplying a column vector X

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & \ldots & a_{1m} \\ a_{21} & a_{22} & & & \vdots \\ \vdots & \vdots & & & \vdots \\ a_{m1} & \ldots & & & a_{mm} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{pmatrix} = \begin{pmatrix} N_1 \\ N_2 \\ \vdots \\ N_m \end{pmatrix}$$

The column vector $P_i = (a_{1i}\ a_{2i}\ \ldots\ a_{mi})$ will represent the i^{th} activity and we have a solution in terms of m activities, some of which may be slack variables.

The costs of activities $P_1, P_2 \ldots P_m$ are $c_1, c_2 \ldots c_m$. If $P = (a_1\ a_2 \ldots a_m$ is an undetermined new activity of cost c, we find a column vector U, such that

$$A.U = P$$

P will improve the current solution if

$$C.U > c$$

$$C.U = CA^{-1}P$$

where $C = (c_1\ c_2 \ldots c_m)$.

We must, therefore, have the row vector CA^{-1} available in our computation.

If we let CA^{-1} be the row vector $(b_1\ b_2 \ldots b_m)$, our conditions that the vector P improves the solution are

$$L_j \geq l_1 a_1 + l_2 a_2 \ldots + l_m a_m$$

and

$$b_1 a_1 + b_2 a_2 + \ldots + b_m a_m > c_{L_j}$$

where c_{L_j} is the cost of the stock length L_j.

This auxiliary problem is solved by the Knapsack Algorithm described in 'A Linear Programming Approach to the Cutting Stock Problem' by Drs. Gilmore and Gomory (Part 1 in J.O.R.S.A 9, 1961, and Part 2 in J.O.R.S.A 11, 1963).

At present, both programs will accept up to 49 different order sizes and up to 15 different stock sizes. They require 16K words of storage (without backing storage), a line printer, and a card or paper-tape reader. The programs are provided as binary dumps on cards, paper tape or the Program Library Tape.

Timing varies approximately as the square of the number of orders. Thus, while a 6-order application requires approximately $\frac{1}{4}$ minute of processing time, a 24-order application requires some 4 minutes.

Chapter 1

INPUT DATA

GENERAL 1.1

The data required by the program is punched either on cards or on paper tape, depending on which of these input media is used for the program. The actual data - order and stock information - must be preceded by a series of parameters.

Data and data sequence are the same on cards and paper tape. Wherever a new card is started with punched card input, a newline character is punched on paper tape.

An example of input data is given in Chapter 4.

PUNCHING PARAMETERS 1.2

Parameters are punched on a single card starting at column 1 or on a length of paper tape terminated by a newline character. They are punched in free format, one space being left between each pair of numbers, in the following sequence:

Parameter	*Significance*
m	Number of different ordered sizes (<50)
k	Number of different stock sizes (<16)
r	Maximum number of pieces that can be cut from a single stock length. If there is no practical restriction, any large number, say 100, may be punched.
z	Number specifying the point at which improvements in the current solution are no longer considered significant. (The solution is considered optimal if all shadow prices are $<z$.) Normally, z is in the range 0.1% to 1.0% of maximum c_j (cost per unit stock length).
n	The maximum number of cycles to occur in the 'dynamic iteration' part of the program. If an improving activity is not found after n cycles, the program proceeds to a final stop. Normally, n is in the range 100 to 300.
u	The minimum acceptable usage for any stock size.

All the parameters except z must be punched as integers.

After reading the input parameters, the program prints a list of them to provide the user with a visual check that they have been punched correctly.

3343(5.66)

PUNCHING DATA

I.3

These are two types of data: order data and stock data. Order data consists of the order lengths required and the quantity of each. Stock data consists of the stock lengths available and their cost per unit length.

Order data is punched first, in any sequence. Each order length is punched as a free format integer starting at column 1 of a new card, or immediately after a newline character on paper tape. Each order length must be followed by a space and another free format integer representing the number of pieces of that length needed to fill the order.

Stock data is then punched in *descending* order of stock lengths. Each available stock length is punched as a free format integer starting at column 1 of a new card, or immediately after a newline character on paper tape. Each stock length must be followed by a space and its cost per unit length punched as a free format number with decimal places if required.

All lengths must be expressed in the same units. If fractional lengths are encountered, all lengths must be converted into multiples of the smallest fraction appearing. If, for example, lengths of $3\frac{1}{4}$" and $1\frac{1}{16}$" were required, sixteenths of an inch would be selected as the unit, and the lengths would be expressed as 52 and 17.

Chapter 2

OUTPUT RESULTS

GENERAL 2.1

As soon as the input parameters and data have been read in, a printout of these is given. This can be used as a visual check on the data and if an error is found the program can be halted and corrections can be made.

The program then calculates an initial solution to the problem and proceeds to improve upon this. Because of the mathematical techniques used in this program, the 'solution' at any stage will consis of the same number of activities as there are orders. An improvement in the solution is the result of finding a new activity to replace one of the current activities. As each improvement is carried out, the program prints the iteration number and current total cost e.g.

ITERATION 21 COST 1120.70

When no further improvement is possible, the program gives a detailed printout of the solution.

PRINTOUT OF SOLUTION 2.2

Each activity which is present in the final solution is printed out separately under the title PATTERN (N). The pattern numbers start at 1 and are consecutive. For each pattern, the stock length used, the number of pieces of stock required, the cost per unit and the total cost are printed on the first line. This is followed by a list of order lengths being cut in the pattern and the number of pieces of each produced from one piece of stock.

The final line of each pattern gives details of the trim or waste incurred. The first figure is the trim on a single piece of stock, i.e. the difference between the stock length and the total length of the ordered pieces being cut. The second figure is the product of the trim and the total number of pieces of stock used. Since this is not an integer, a further figure, the product of the trim and the rounded up value of the number of pieces of stock used, is also given.

When all the patterns have been printed, a summary is given of the rounded and unrounded cost and the rounded and unrounded waste, together with value of the percentage unrounded waste and percentage rounded waste. This is followed by a list of the stocks selected by the program and the quantities used.

Finally, the message OUTPUT COMPLETED is printed.

Chapter 3

OPERATING INSTRUCTIONS

This chapter gives a brief summary of the operating instructions for running the Trim Loss programs, #XUT1 and #XUT2. The instructions given are for program #XUT1; they are the same for #XUT2 except for the program name used in the typed messages.

BINARY DUMP PROGRAM ON PAPER TAPE 3.1

1 Type the following message on the console typewriter:

LO #XUT1

2 Continue with step 2 of section 3.2.

PROGRAM ON MAGNETIC TAPE 3.2

1 To load the program from the Scientific Library tape into core store, type the following message on the console typewriter:

FI #XUT1

2 Load the data on the card or paper tape reader as appropriate. If the data is on cards, type the message

ON #XUT1 1

If the data is on paper tape, no action is necessary at this stage.

3 To activate the program, type the message

GO #XUT1

4 At the end of the run, the program outputs the message

O #XUT1; HALTED:- 00

5 To process further data, load the new data and return to step 3.

SPECIAL INSTRUCTION FOR A SHORT RUN 3.3

The programs print the current cost and iteration number each time a new pattern is introduced into the solution. If the time taken by the run exceeds the available time without converging to the specified cut off, a printout of the current solution can be obtained and the run terminated, by typing the message

ON #XUT1 2

Chapter 4

EXAMPLE

The following printout is the result of applying a 1900 Trim Loss program to a problem in which three stock lengths are available, and various quantities of six order lengths are required.

The input list on the first sheet illustrates the stock and order information, shows that only four knives are available for any given cutting activity, and indicates that a given stock length is to be used only if at least 12000 order lengths are to be cut from pieces of that length.

Six iterations take place, the cost being decreased from 155961.45 to 149631.18.

After the sixth iteration, the cutting patterns are printed, there being one pattern or activity to correspond to each order length.

The summary on the final sheet shows that the 1296 stock length is not used at all, and that the suggested cutting patterns result in a wastage of only 2.34%.

3343(5.66)

STOCK

	LENGTH	COST
1	1344	3.6020
2	1296	3.5140
3	1200	3.2570

ORDERS

	LENGTH	NUMBER
1	573	37862
2	384	30388
3	396	22500
4	352	25580
5	415	3250
6	725	2346

MAX NO OF PIECES CUT FROM ONE STOCK LENGTH	5
DYNAMIC ITERATION LIMIT	200
CUT-OFF POINT	.005
MINIMUM USAGE OF STOCK LENGTH	12000

ITERATION	1	COST	155961.45
ITERATION	2	COST	155574.68
ITERATION	3	COST	155470.72
ITERATION	4	COST	152985.15
ITERATION	5	COST	152637.06
ITERATION	6	COST	149631.18

PATTERN 1

STOCK LENGTH	NUMBER USED	COST	TOTAL COST
1200	4172.2	3.257	13589.02

ORDER LENGTH	NUMBER CUT
573	2

TRIM	UNROUNDED WASTE	ROUNDED WASTE
54	225301.5	225342.0

PATTERN 2

STOCK LENGTH	NUMBER USED	COST	TOTAL COST
1344	15194.0	3.602	54728.79

ORDER LENGTH	NUMBER CUT
573	1
384	2

TRIM	UNROUNDED WASTE	ROUNDED WASTE
3	45582.0	45582.0

PATTERN 3

STOCK LENGTH	NUMBER USED	COST	TOTAL COST
1200	7500.0	3.257	24427.50

ORDER LENGTH	NUMBER CUT
396	3

TRIM	UNROUNDED WASTE	ROUNDED WASTE
12	90000.0	90000.0

PATTERN 4

STOCK LENGTH	NUMBER USED	COST	TOTAL COST
1344	11977.5	3.602	43142.95

ORDER LENGTH	NUMBER CUT
573	1
352	2

TRIM	UNROUNDED WASTE	ROUNDED WASTE
67	802492.5	802526.0

PATTERN 5

STOCK LENGTH	NUMBER USED	COST	TOTAL COST
1200	1625.0	3.257	5292.63

ORDER LENGTH	NUMBER CUT
352	1
415	2

TRIM	UNROUNDED WASTE	ROUNDED WASTE
18	29250.0	29250.0

PATTERN 6

STOCK LENGTH	NUMBER USED	COST	TOTAL COST
1344	2346.0	3.602	8450.29

ORDER LENGTH	NUMBER CUT
573	1
725	1

TRIM	UNROUNDED WASTE	ROUNDED WASTE
46	107916.0	107916.0

SUMMARY

	TOTAL COST		TOTAL WASTE	
	UNROUNDED	ROUNDED	UNROUNDED	ROUNDED
	149631.2	149632	1300542.0	1300616
PERCENTAGE WASTE			2.34	2.34

STOCK

	LENGTH	COST	NO.USED
1	1344	3.6020	29518.
2	1200	3.2570	13298.

OUTPUT COMPLETED

Amendment record

Should any additions or amendments to this Manual be necessary, they will be issued in the form of Amendment Lists.
When such changes have been incorporated, an appropriate entry should be made in the table below and the pink instruction sheet(s) should be filed behind this page.

List number	Item number
	to
	to
	to
	to
	to
	to
	to
	to
	to
	to
	to
	to
	to
	to
	to
	to
	to
	to
	to
	to
	to
	to
	to
	to
	to
	to
	to
	to
	to
	to
	to
	to
	to
	to