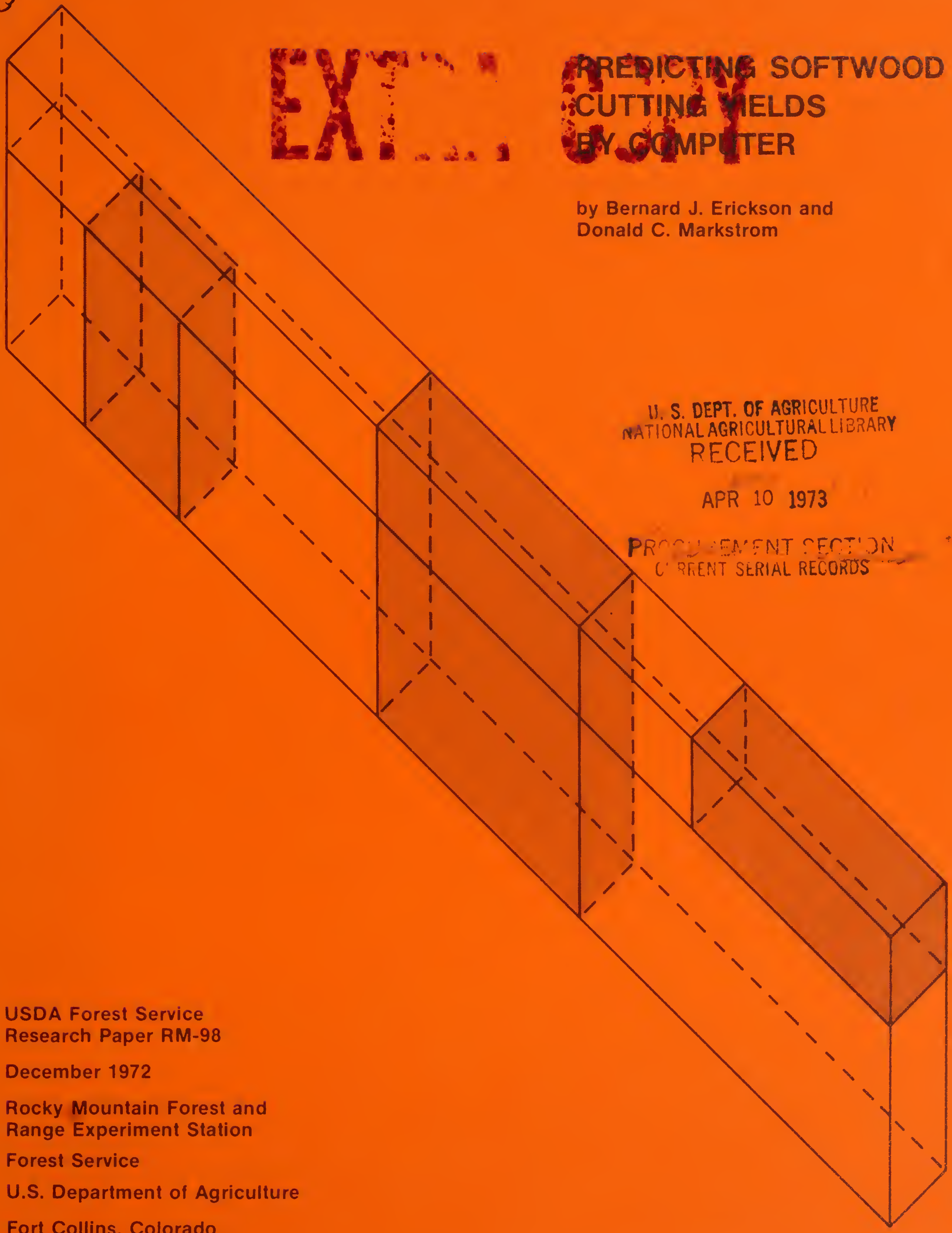
PREDICTING SOFTWOOD CUTTING YIELDS BY COMPUTER

by Bernard J. Erickson and
Donald C. Markstrom



USDA Forest Service
Research Paper RM-98

December 1972

Rocky Mountain Forest and
Range Experiment Station

Forest Service

U.S. Department of Agriculture

Fort Collins, Colorado

Preface and Acknowledgments

The computer program presented here was developed in a peripheral study in support of an analysis of the feasibility of roll laminating fiber overlays on low-grade ponderosa pine lumber. A part of that study involved determining economically and technically feasible methods for eliminating open defects in lumber to be used as overlay substrates. The overall study, done in cooperation with the Duke City Lumber Company, Albuquerque, New Mexico, and financed by the Economic Development Administration, U.S. Department of Commerce, is reported in USDA Forest Service Research Paper RM-97.

Several people, in addition to the authors, contributed to the development of the program. Arthur Bourke and John McElfresh, consultants to the Rocky Mountain Station on the roll laminating study, participated both in formulating the problem and writing the initial program. Hiram Hallock, wood technologist with the USDA Forest Products Laboratory, helped formulate the problem. Jacob Kovner, biometrician, and Lincoln Mueller and Roland Barger, wood technologists, Rocky Mountain Station, also provided guidance during the course of the study. Barger is now at the Intermountain Forest and Range Experiment Station, and Mueller is retired.

Abstract

A computer program, written in FORTRAN predicts the maximum yield of cuttings for softwood cut-up and edge- and end-gluing operation. The program calculates cutting recovery (given cutting width and length constraints and defect locations on the board), and locates ripping saw kerfs.

Keywords: Programming (computers), softwood cut-up, end and edge gluing, dimension stock.

USDA Forest Service
Research Paper RM-98

December 1972

Predicting Softwood Cutting Yield by Computer

by

Bernard J. Erickson, Computer Programer,

and

Donald C. Markstrom, Associate Wood Technologist

Rocky Mountain Forest and Range Experiment Station[1]

[1] *Central headquarters maintained at Fort Collins, in cooperation with Colorado State University.*

Contents

Page

Predicting Softwood Cutting Yield by Computer

Bernard J. Erickson and Donald C. Markstrom

Program CUTUP in Brief

Currently, major emphasis in improving wood conversion efficiency is concentrated in three areas: (1) improving yields from the resource, (2) reducing waste and residues, and (3) reducing processing errors resulting from repetitive human decisions. The program described here can contribute to greater efficiency in all of these areas. Automated defect-sensing methods now being developed offer the possibility of feeding continuous defect data to a computer as input to the type of optimizing program described, thereby automating the whole cut-up decision process.

A computer program CUTUP to predict the maximum yield of cuttings obtainable from softwood lumber is described in this paper. The program calculates cutting recovery (given cutting width and length constraints and defect location on the board), and will predict the maximum yield of cuttings of a specified grade, for a softwood cut-up and edge- and end-gluing operation.

The conventional practice in softwood cut-up operations is to first rip the board into prescribed cutting widths in a manner to maximize recovery, and then crosscut the full-board-length cuttings to remove defects. The CUTUP computer program simulates this practice systematically, (1) locating full-board-length cuttings of specified widths and sequence to maximize recovery, (2) crosscutting to remove defects, and (3) measuring the areas of all cuttings of a specified length or longer. The recovered area excludes (1) the area of defects, (2) areas between defects, or between a defect and the end of the board, where the recoverable length is less than the specified minimum, and (3) areas of all saw kerfs. The maximum number of rips is also specified in the program.

This program differs fundamentally from computer programs developed to predict yield of cuttings from hardwood cut-stock operations, where the usual sawing practice is to alternately crosscut and rip.[2] [3] Hardwood programs systematically scan the board to locate and measure cuttings of prescribed width and length between defect areas, with the largest specified cutting considered first. The cuttings can then be removed from the board by starting with either a crosscut or rip, and alternating back and forth, taking as many cuts as possible during each operation. Softwood and hardwood programs are similar, however, in that the same format is used to record board and defect data on the input cards.

Characterizing Boards and Defects

Each board is considered to have X and Y axes, with its lower left corner at (0,0), its length along the X axis, and its width along the Y axis. The board size is described and defects are located by 1/4-inch coordinates, which designate the lower left and upper right corners of the areas (fig. 1). All measurements are expressed in 1/4-inch units, which can be used directly in calculating recovery of cut-up stock.

Data cards describing the boards and defects are punched in the following manner: The first card lists the board grade code in columns 6 and 7, the board number in columns 32 through 36, and total number of defects in columns 69 through 72. The second card gives the coordinates of the board, and each of the remaining cards gives the coordinates of a single defect. The lower left Y coordinates of the board and the defects are punched in columns 1 through 3 and the X coordinates in columns 5 through 7. The upper right Y coordinates are punched in columns 13 through 15 and the X coordinates in columns 17 through 19. The sequence of the defect cards in the deck must be the same as that of the defects, from left to right across the X axis of the board.

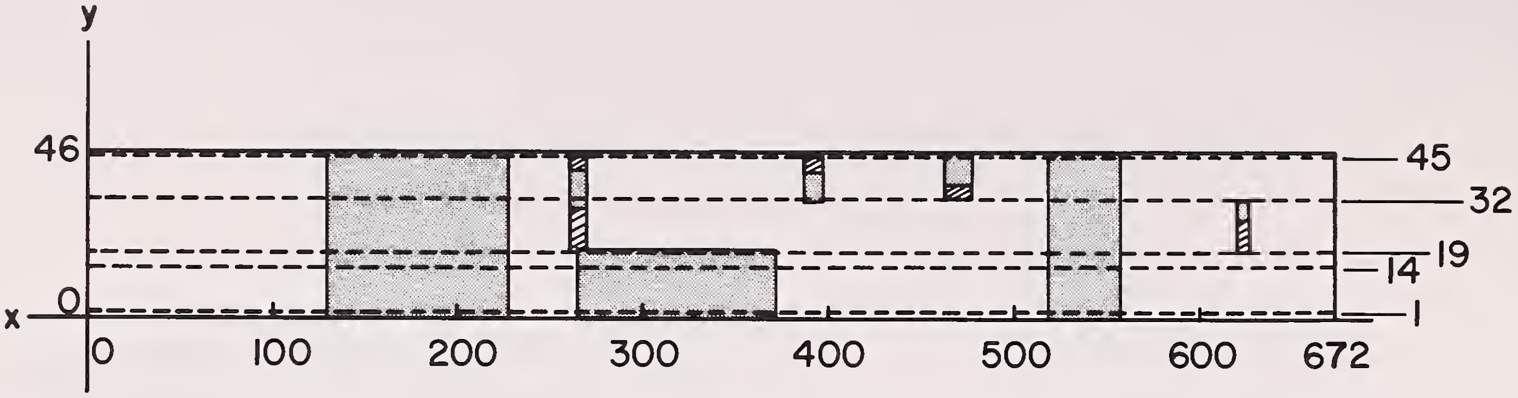
The total number of cards for each board equals the number of defects plus two.

The program was run with a sample of 600 boards selected from surfaced yard lumber at the Duke City Lumber Company. One hundred boards were randomly selected in each of six categories: grades 3 Common and 4 Common, in nominal widths of 4, 8, and 12 inches.

The boards were examined and defects outlined. As a convenience, the boards were then photographed with color transparency film. The slides were projected at one-half scale on rear-projection translucent glass to facilitate measuring board and defect data.

[2] *Englerth, G. H., and D.E. Dunmire. Programing for lumber yield. For. Prod. J. 16(9): 67-69. 1966.*

[3] *Wodzinski, Claudia, and Eldona Hahm. A computer program to determine yields of lumber. U.S. Dep. Agric., For. Prod. Lab., 33 p. Madison, Wis. 1966.*



```
Board grade:  4C                                          Defect coordinates: 1.  (0,127)  (46,222)
      width:  11½ inches (46 units)
     length:  14 feet (672 units)                                             2.  (29,259)  (38,265)
coordinates:  (0,0) (46,672)
                                                                              3.  (0,262)  (19,370)
Cutting widths:  1, 2, and 3 inches
        length:  9 inches +                                                   4.  (33,386)  (41,396)

Optimum cutting widths and sequence:  3, 1, 3, and 3 inches                   5.  (35,463)  (46,475)
Saw kerf coordinates:                 1, 14, 19, 32, 45, and 46
Area recovered:                       19,404 sq. units                        6.  (0,519)  (46,556)
Area of board:                        30,912 sq. units
Percent recovery:                     62.77 percent                           7.  (27,620)  (31,625)
```

Figure 1.--Diagrammatic sketch of a grade 4 Common 1- by 12-inch board, with associated program output data, illustrates the program developed for selective cut-up evaluation. Shaded areas represent defects that have been located by X-Y coordinates and blocked out. Lined areas represent additional material lost in cutting out defects.

Using the Program

Boards of different species, sizes, and grades could be sampled and the program used to evaluate the following:

1. Area recovered using alternative rip combinations.
2. Efficiency (in percent recovery) obtained from alternative widths of boards, for specified rip widths.
3. Effects of grade on rip recovery, for given species and specified rip widths.
4. Effects of adding or deleting rip widths from the array now used.
5. Alternative costs of input stock of various species, grades, and widths, based on predicted potential.

The program can undoubtedly be used to evaluate many other situations.

Program CUTUP

Program CUTUP is a set of three main programs — PERMU, CONST, and RECVRY. Program PERMU prepares a table of all possible cutting combinations for the N cutting widths (with M saw blades) to be considered for a given board-width class. User-selected cutting combinations are input to program RECVRY, which fits the cuts to individual boards to obtain maximum recovery area. Program CONST has one subroutine, XOUT, which outlines or "x's out" the defective areas of a board by relating defect, X, Y coordinates to computer core storage locations. The program prepares an input file of nonrecoverable board-defect areas, by specified cutting widths, for use in the RECVRY program. For boards 7.5 inches and wider, CONST requires considerable computer time. If narrower widths are to be processed, programs CONST and RECVRY may be run together as one program, with the output from CONST used directly as input to RECVRY. This eliminates the necessity of saving CONST output (Tape) for further processing by program RECVRY.

The board-defect data cards which are input to program CONST should be edited and error checked before use. If a large amount of data is to be processed, the data should be tape filed for ease of handling. A sample edit and card-to-tape program is included with the program listings in appendix 2.

The basic content and purpose of each program and subroutine is described in the sections that follow. The order and format of input control cards for each program are also specified. Variable names are defined in the source listings for each program. A sample output of an application of program CUTUP is shown in appendix 1, which also further explains the overall program operation.

Program PERMU, CONST, and RECVRY were originally written in the BASIC programming language for use on a time-sharing computer system. The programs were modified, expanded, and rewritten in FORTRAN extended for use on a Control Data Corp. 6400 computer,[4] and have been tested on the CDC 6400 computer system at the Colorado State University Computing Center.

Program PERMU

This program produces a table of possible cutting combinations for a specified board width, cutting or ripping widths, and cutting positions (kerfs). The computational procedure is based upon the principle that if one event A can take place in m ways, and a second event B can take place in n independent ways, then the number of ways in which both events can take place is mn. This principle can be extended to any number of independent events. The procedure was applied to the problem of cutting boards into different widths where each saw kerf is an event which can happen in m positions of the saw blade over a board. By expanding this technique, all combinations previously derived for narrower cutting widths and fewer cutting blades are saved for possible application in dealing with wider cutting widths and more cutting blades.

The procedure was used to process data involving three cutting widths of four, eight, and twelve 1/4-inch units (Q.I.U.), and five kerfs or cutting positions. Sample board widths ranged from 14 to 46 Q.I.U. To accommodate additional cutting widths and/or cutting positions, test parameters and array dimensions would need to be changed, as specified by the analysis problem.

Program Control Card Type 1. FORMAT (315)

Column		
1-5	IBWDT	Overall board width in 1/4-inch units (Q.I.U.)
6-10	NSIZE	Number of different cutting sizes or rip widths to consider, MAX = 5.
11-15	NSWBL	Number of machine saw blades available, MAX = 6.

Program Control Card Type 2. FORMAT (515)

Column		
1- 5	KUT(1)	Size of cutting width 1, in Q.I.U.
6-10	KUT(2)	Size of cutting width 2, in Q.I.U.
11-15	KUT(3)	Size of cutting width 3, in Q.I.U.
16-20	KUT(4)	Size of cutting width 4, in Q.I.U.
21-25	KUT(5)	Size of cutting width 5, in Q.I.U.

Program CONST

This program constructs, in computer core storage, a board of variable width and length dimensions, and positions the location of defects in the board according to their coordinates. Board and defect coordinates are expressed in Q.I.U. to the X, Y coordinate system.

The amount of core storage available in the computer determines the size of boards which can be analyzed by this program. The array LWB(I,J), **L**ength and **W**idth of **B**oard, is dimensioned according to the length (J) and width (I) of the largest boards to be analyzed in Q.I. U. If the largest boards to be processed are 11.5 inches by 16 feet, the array LWB(I,J) would be dimensioned I =46, J = 768, resulting in 35,328 computer core locations to store all possible locations of defects.

Input data are contained in the set of cards which specify the board and defect coordinate values for one board (described in Characterizing Boards and Defects). The defects are positioned in LWB according to increasing defect coordinates in the J, or length, dimension of LWB(I,J).

If a large amount of data is to be processed, and is arranged or sorted by some criteria such as increasing board numbers within a board-width class, an option is provided to analyze a selected sub-set from the complete file. This option is exercised by two control parameters labeled NBSTR and NBEND, where NBSTR is the number of the first board in the sub-set and NBEND is the number of the last board in the sub-set to be processed.

A user input parameter labeled LBTD is a test variable used for setting a minimum acceptable length between defects. If an area between two defects (or between a defect and the end or start of a board) is less than LBTD (in Q.I.U.), the area is blocked out and treated as a defect area in the final scan of the board to locate all defect areas.

For each input board, program CONST operates in the following manner:

1. The board ID or Header Card is read, setting the variables NGRD, (board grade class), NBRD (board number), and NDEF (number of defects in board).
2. A board coordinate card is read **defining** the X-dimension (Width) as the ordinate and the Y-dimension (Length) the abscissa of the board to be processed. This notation is

[4] *The use of trade and company names is for the benefit of the reader; such use does not constitute an official endorsement or approval of any service or product by the U. S. Department of Agriculture to the exclusion of others that may be suitable.*

reversed from the normal X, Y notation because of computer core storage procedure.

3. A set of defect coordinate cards is read until an END-OF-BOARD data card is encountered (IDXS=999).

The coordinates for each defect are positioned in an array by a pointer according to their order of occurrence within the board. Using the minimum length between defects test (LBTD), the defect array is scanned to find areas between defects which are less than LBTD. Defect location variables NX, LDS, and LDE are determined and passed to subroutine XOUT, along with the board array LWB, by the use of a COMMON statement. Subroutine XOUT then blocks out or "x's-out" the defined area of the defect in the board. This process continues until all defect areas have been positioned in the LWB array.

Indexes are now determined for a scan of the board to accumulate total defect area for NSIZE desired cuttings over a range of possible starting cut positions determined by board width. The total defect area for the board is written to an output file (Disk or Tape) for input to program RECVRY. The next board ID or Header Card is read and the process is repeated until an end-of-file is encountered in the board input data.

Program Control Card. FORMAT (3I5)

Column		
1- 5	LBTD	Minimum length between defects limitation (Q.I.U.)
6-10	NBSTR	Board number to start analysis
11-15	NBEND	Board number to end analysis

Board Defect Data Card Deck

NDEF + two cards for each board to be processed plus an END-OF-BOARD Card.

Program RECVRY

This program computes the nondefective areas in a board, and simulates a number of possible cuttings, limited by the width of the board and the combinations of cutting sizes desired, to determine the best combination to use for maximum recovery.

Input for the program is the board-defect area file (Tape or Disk) from program CONST, and a user-selected set of cutting combinations generated by program PERMU to be evaluated. The cutting combinations are input to the program in a DATA statement, array KPCT. The coding order is explained in the program listing.

The program reads a title card and an input parameter control card. The parameter LPRM is the lower limit of combined cuttings to be evaluated from the combination sets in array KPCT. Only combined cuttings greater than LPRM are used to evaluate the best fit, thus eliminating cutting combinations which do not use the entire width of the board. The variable NSIZE and the array KUT are set from values coded in DATA KSZE.

The board ID and board coordinates are read from the input file, along with board defect areas. Areas of nondefect are computed from defect area and area of the board, and saved. Indexes are set for selecting cutting combinations from array KPCT to obtain maximum recovery for the board. The number of saw kerfs which can be placed within the bounds of the board is determined. A saw kerf may be adjacent to the lower or upper edges of a board, or it may lie within the board itself. There is no provision in this program for straightening the edges and ends of the board to assure a straight board. A warped board may be utilized, however, by crosscutting into shorter lengths or ripping into narrow widths, by assigning appropriate defect coordinates. The sum of the cutting widths and saw kerfs cannot exceed the board width plus one unit.

A total recovery area is computed from the nondefective area data for a selected combination of cuttings. This value is stored as the "best" until a larger recovery area is obtained from a different combination of cuts, which then replaces it as the "best." After all acceptable combinations have been evaluated for best fit, the individual board recovery summary is printed out. Totals are accumulated for total recovery output by board grade and size class until an end-of-file is encountered in the input data file.

Program Control Card Type 1. FORMAT (4A10)

Column		
1-40	TITLE	Short title to identify data being processed.

Program Control Card Type 2. FORMAT (4I5)

Column		
1- 5	LBTD	Minimum length between defects control: same value as in CONST.
6-10	LPRM	Lower limit of combined cuttings control: used to select combinations of cutting sizes.
11-15	NBSTR	Board number to start analysis for recoverable area.
16-20	NBEND	Board number to end analysis for recoverable area.

Board numbers (NBRD) to be included in the analysis should be ≥NBSTR and ≤NBEND.

Appendix 1: Sample Output From Programs

The sample output following is a set of computer printout pages obtained from running the PERMU and CONST/RECVRY programs. The board input data consisted of a set of 10 boards from the 14 Q.I.U. (3.5-inch) board width class. Five boards, numbered 291-295, were grade 3 Common with a length of 768 Q.I.U. (16 feet), and five boards, numbered 301-305, were grade 4 Common with a length of 576 Q.I.U. (12 feet). The minimum acceptable length between defects was set at 36 Q.I.U. (9 inches). Only those cutting combinations which resulted in a total cut (width) greater than 8 Q.I.U. and less than or equal to 14 Q.I.U. (total board width) were evaluated. Cutting combinations evaluated included (numbers in Q.I.U.), (4, 8), (8, 4), (12). The only other possible combination (4, 4, 4) was excluded to decrease the number of 1-inch-wide cuttings.

PERMU Output

The first line is a summary of the input values assigned to the program control parameters IBWDT, NSIZE, NSWBL, and the array KUT.

IBWDT	= 46 Q.I.U.	Maximum width of sample boards, 11.5-inches
NSIZE	= 3	Number of cutting sizes to consider
NSWBL	= 4	Number of machine saw blades available
KUT(1)	= 4 Q.I.U.	Consider a 1-inch cutting
KUT(2)	= 8 Q.I.U.	Consider a 2-inch cutting
KUT(3)	= 12 Q.I.U.	Consider a 3-inch cutting
KUT(4)	= 0	Not used this run
KUT(5)	= 0	Not used this run

The total number of possible combinations for three sizes and four blades is the computed value 120. Only those combinations which can be made using four positions within the limits of the board are printed out for user consideration.

Column 1 of the output table is a combination (COMBO) number used to reference the cutting combination. Columns 2-7 contain cutting combinations. The number 6 is used to fill out the table, and has no significance as far as combinations are concerned. The numbers 1, 2, and 3 in the table are referenced to the cutting sizes stored in array KUT so that 1 = 4, 2 = 8, and 3 = 12 Q.I.U. The number in column 8 is the total units cut for the given cutting combination and is used as a test limit in program RECVRY. The table output is ordered by increasing values in column 8.

```
 46     3     4     4     8    12     0     0



NBR OF DIFFERENT CUT COMBINATIONS =     120

  1     1     6     6     6     6     6     4
  2     2     6     6     6     6     6     8
  4     1     1     6     6     6     6     8
  3     3     6     6     6     6     6    12
  5     2     1     6     6     6     6    12
  7     1     2     6     6     6     6    12
 13     1     1     1     6     6     6    12
  6     3     1     6     6     6     6    16
  8     2     2     6     6     6     6    16
 10     1     3     6     6     6     6    16
 14     2     1     1     6     6     6    16
 16     1     2     1     6     6     6    16
 22     1     1     2     6     6     6    16
 40     1     1     1     1     6     6    16
  9     3     2     6     6     6     6    20
 11     2     3     6     6     6     6    20
 15     3     1     1     6     6     6    20
 17     2     2     1     6     6     6    20
 19     1     3     1     6     6     6    20
 23     2     1     2     6     6     6    20
 25     1     2     2     6     6     6    20
 31     1     1     3     6     6     6    20
 41     2     1     1     1     6     6    20

 43     1     2     1     1     6     6    20
 49     1     1     2     1     6     6    20
 67     1     1     1     2     6     6    20
 12     3     3     6     6     6     6    24
 18     3     2     1     6     6     6    24
 20     2     3     1     6     6     6    24
 24     3     1     2     6     6     6    24
 26     2     2     2     6     6     6    24
 28     1     3     2     6     6     6    24
 32     2     1     3     6     6     6    24
 34     1     2     3     6     6     6    24
 42     3     1     1     1     6     6    24
 44     2     2     1     1     6     6    24
 46     1     3     1     1     6     6    24
 50     2     1     2     1     6     6    24
 52     1     2     2     1     6     6    24
 58     1     1     3     1     6     6    24
 68     2     1     1     2     6     6    24
 70     1     2     1     2     6     6    24
 76     1     1     2     2     6     6    24
 94     1     1     1     3     6     6    24
 21     3     3     1     6     6     6    28
 27     3     2     2     6     6     6    28

 29     2     3     2     6     6     6    28
 33     3     1     3     6     6     6    28
 35     2     2     3     6     6     6    28
 37     1     3     3     6     6     6    28
 45     3     2     1     1     6     6    28
 47     2     3     1     1     6     6    28
 51     3     1     2     1     6     6    28
 53     2     2     2     1     6     6    28
 55   1   3   2   1   6   6   28
 59   2   1   3   1   6   6   28
 61   1   2   3   1   6   6   28
 69   3   1   1   2   6   6   28
 71   2   2   1   2   6   6   28
 73   1   3   1   2   6   6   28
 77   2   1   2   2   6   6   28
 79   1   2   2   2   6   6   28
 85   1   1   3   2   6   6   28
 95   2   1   1   3   6   6   28
 97   1   2   1   3   6   6   28
103   1   1   2   3   6   6   28
 30   3   3   2   6   6   6   32
 36   3   2   3   6   6   6   32
 38   2   3   3   6   6   6   32

 48   3   3   1   1   6   6   32
 54   3   2   2   1   6   6   32
 56   2   3   2   1   6   6   32
 60   3   1   3   1   6   6   32
 62   2   2   3   1   6   6   32
 64   1   3   3   1   6   6   32
 72   3   2   1   2   6   6   32
 74   2   3   1   2   6   6   32
 78   3   1   2   2   6   6   32
 80   2   2   2   2   6   6   32
 82   1   3   2   2   6   6   32
 86   2   1   3   2   6   6   32
 88   1   2   3   2   6   6   32
 96   3   1   1   3   6   6   32
 98   2   2   1   3   6   6   32
100   1   3   1   3   6   6   32
104   2   1   2   3   6   6   32
106   1   2   2   3   6   6   32
112   1   1   3   3   6   6   32
 39   3   3   3   6   6   6   36
 57   3   3   2   1   6   6   36
 63   3   2   3   1   6   6   36
 65   2   3   3   1   6   6   36

 75   3   3   1   2   6   6   36
 81   3   2   2   2   6   6   36
 83   2   3   2   2   6   6   36
 87   3   1   3   2   6   6   36
 89   2   2   3   2   6   6   36
 91   1   3   3   2   6   6   36
 99   3   2   1   3   6   6   36
101   2   3   1   3   6   6   36
105   3   1   2   3   6   6   36
107   2   2   2   3   6   6   36
109   1   3   2   3   6   6   36
113   2   1   3   3   6   6   36
115   1   2   3   3   6   6   36
 66   3   3   3   1   6   6   40
 84   3   3   2   2   6   6   40
 90   3   2   3   2   6   6   40
 92   2   3   3   2   6   6   40
102   3   3   1   3   6   6   40
108   3   2   2   3   6   6   40
110   2   3   2   3   6   6   40
114   3   1   3   3   6   6   40
116   2   2   3   3   6   6   40
118   1   3   3   3   6   6   40
```

CONST Output

The first line is a statement of the input values assigned to the program control parameters LBTD = 36, NBSTR = 291, and NBEND = 305. The next 10 lines are the input data for the first board, followed by the computed defect area for three cutting sizes for that board. Complete output for all the test boards is printed out for the sample run. When a large number of boards is to be evaluated, complete output, for only the first board in the input set is printed to reduce the number of output pages required. The remaining boards in the input set are indicated only by their ID, which contains the grade class, board number, number of defects in the board, and the overall coordinates of the board. The last line is a statement indicating the run was successfully completed, and gives the total number of boards processed.

```
MIN. LENGTH BETWEEN DEFECTS CONTROL FOR
THIS RUN IS    36    291 -   305

 3 291           8
 0   0          14 768
11 366          14 374
 0 411           7 421
 0 444           1 473
 4 451          12 463
 0 537           2 547
 0 574           3 582
 8 667          13 671
 3 697          13 710
  1       480     960     1896
  2       440    1120     1776
  3       292     824     1332
  4       260     760        0
  5       260     824        0
  6       380     824        0
  7       380     824        0
  8       220       0        0
  9       252       0        0
 10       252       0        0
 11       252       0        0
 12         0       0        0
 13         0       0        0
 14         0       0        0

 3 292           5
 0   0          14 768
 0 208           9 232
 4 299          13 316
 0 514           4 525
 1 533          10 550
 5 637          10 642
  1       240     656      984
  2       308     656      984
  3       328     656      984
  4       328     656        0
  5       252     504        0
  6     252     504       0
  7     252     504       0
  8     252       0       0
  9     252       0       0
 10     156       0       0
 11      68       0       0
 12       0       0       0
 13       0       0       0
 14       0       0       0

 3 293         7
 0   0        14 768
 0  31         3  37
 8 113        14 123
 2 508         7 514
 1 525         5 530
 0 543        14 544
 9 609        14 618
10 711        14 716
  1     292     584    1164
  2     292     664    1164
  3     292     736    1164
  4     144     480       0
  5     144     480       0
  6     184     480       0
  7     220     480       0
  8     100       0       0
  9     100       0       0
 10     100       0       0
 11     100       0       0
 12       0       0       0
 13       0       0       0
 14       0       0       0

 3 294         5
 0   0        14 768
10 245        14 252
 0 295        14 355
10 359        14 381
 9 646        14 658
 9 708        14 717
  1     240     480    1368
  2     240     480    1368
  3     240     648    1368
  4     240     912       0
  5     240     912       0
  6     240     912       0
  7     324     912       0
  8     456       0       0
  9     456       0       0
 10     456       0       0
 11     456       0       0
 12       0       0       0
 13       0       0       0
 14       0       0       0

 3 295         7
 0   0        14 768
11  75        14  84
 0 175         4 190
 5 431        12 439
 6 545        13 554
 8 661        14 669
 0 679         5 689
 0 737        10 768
```

```
  1     224     584    1200
  2     224     728    1200
  3     256     728    1200
  4     292     728       0
  5     232     680       0
  6     224     520       0
  7     224     520       0
  8     224       0       0
  9     260       0       0
 10     260       0       0
 11     136       0       0
 12       0       0       0
 13       0       0       0
 14       0       0       0

 4 301         7
 0   0        14 576
11   0        14  26
 2  95         7 100
 0 146         5 153
 2 181         9 186
 7 322        11 327
 0 498         2 576
 3 547         9 555
  1     492    1024    1848
  2     492    1024    1848
  3     296     632    1260
  4     296     632       0
  5     316     840       0
  6     176     560       0
  7     176     560       0
  8     156       0       0
  9     260       0       0
 10     124       0       0
 11     124       0       0
 12       0       0       0
 13       0       0       0
 14       0       0       0

 4 302        12
 0   0        14 576
 0  40         5  47
 0 226         4 237
10 281        14 293
 3 291         9 298
12 307        14 346
 9 348        14 361
 9 384        14 391
 0 438         6 448
 6 475         9 479
 0 505         7 512
 0 544         6 576
12 551        14 576
  1     424    1304    2592
  2     424    1304    3192
  3     424    1648    3192
  4     652    1728       0
  5     608    1640       0
  6     580    1984       0
  7     348    1376       0
  8     256       0       0
  9     256       0       0
 10     540       0       0
 11     540       0       0
 12       0       0       0
```

```
13      0       0       0
14      0       0       0

 4 303       4
 0   0      14 576
 2 161       7 167
 0 290       3 329
 6 420      14 476
 3 502      11 514
 1    228    1112    1668
 2    228    1112    1668
 3    228    1112    1668
 4    400     800       0
 5    400     800       0
 6    400     800       0
 7    400     800       0
 8    376       0       0
 9    376       0       0
10    376       0       0
11    376       0       0
12      0       0       0
13      0       0       0
14      0       0       0

 4 304       8
 0   0      14 576
 9  84      14  93
 7 124      13 132
 0 208       9 222
 9 236      14 246
 0 305       3 312
 4 346      13 358
11 466      14 472
 6 511      14 522
 1     84     688    1872
 2    268     688    1872
 3    268    1200    1872
 4    148     872       0
 5    180     920       0
 6    180     920       0
 7    436     920       0
 8    436       0       0
 9    460       0       0
10    348       0       0
11    348       0       0
12      0       0       0
13      0       0       0
14      0       0       0

 4 305       5
 0   0      14 576
 1  48       5  54
 6 156      11 165
11 320      14 328
11 491      14 498
 0 517       3 576
 1    260     592    1296
 2    260     592    1296
 3    260     592    1296
 4     60     120       0
 5     60     240       0
 6     36     192       0
 7     36     192       0
 8     36       0       0
 9     96       0       0
10     96       0       0
11     96       0       0
12      0       0       0
13      0       0       0
14      0       0       0

END-OF-RUN. TOTAL BOARDS PROCESSED =  10
```

RECVRY Output

The first line is a short title to describe the type of boards processed by the program, and a summary of input values assigned to the program control parameters LBTD = 36, LPRM = 8, NBSTR = 291, and NBEND = 305. Output consists of percent recovery for each individual board, percent total recovery for all boards, percent total recovery for grade 3 Common boards, percent total recovery for grade 4 Common boards, and a summary of the number of times each available cutting combination was selected for the individual boards.

Output for the first board, number 291, shows that combination number 5 generated the "best" recovery area. From the PERMU output sample, combination number 5 is a (2, 1) cutting or an 8 and 4 Q.I.U. cut. Next is a set of six numbers which position the saw kerfs for combination number 5. The first saw kerf is located at unit 0 of the board. Units 1 through 8 are used for the first cut (8 Q.I.U.). The second saw kerf is located at unit 9. Units 10, 11, 12, and 13 are used for the second cut (4 Q.I.U.) and the third saw kerf is located at unit 14. A saw kerf location of zero indicates a position adjacent to the bottom edge of the board. If the saw kerfs and cutting widths do not account for all available units in the board as shown by boards 292, 295, and 302, two adjacent saw kerfs of one unit each are indicated, and only the upper saw kerf location is printed. The total clear area recovered is 8,004 Q.I.U., the total area of the board is 10,752 Q.I.U. (14 x 768), and the percent recovery for the board is 74,442 (8,004/10,752 x 100). This format is repeated for each board in the input file.

```
TEST 4-INCH 10 BRDS MIN LGT=36   11/27/72
36    8  291   305


BOARD NBR  =   291
COMBO NBR  =     5
   0    9   14   15   16   17
AREA RECOVERED =      8004
AREA OF  BOARD =     10752
PRCNT RECOVERY =  74.4420

BOARD NBR  =   292
COMBO NBR  =     5
```

```
   0  10  15  16  17  18
AREA RECOVERED =    8492
AREA OF  BOARD =   10752
PRCNT RECOVERY = 78.9807

BOARD NBR =  293
COMBO NBR =    5
   0   9  14  15  16  17
AREA RECOVERED =    8532
AREA OF  BOARD =   10752
PRCNT RECOVERY = 79.3527

BOARD NBR =  294
COMBO NBR =    5
   0   9  14  15  16  17
AREA RECOVERED =    8280
AREA OF  BOARD =   10752
PRCNT RECOVERY = 77.0089

BOARD NBR =  295
COMBO NBR =    5
   0  10  15  16  17  18
AREA RECOVERED =    8496
AREA OF  BOARD =   10752
PRCNT RECOVERY = 79.0179

BOARD NBR =  301
COMBO NBR =    7
   0   5  14  15  16  17
AREA RECOVERED =    5860
AREA OF  BOARD =    8064
PRCNT RECOVERY = 72.6687

BOARD NBR =  302
COMBO NBR =    7
   0   6  15  16  17  18
AREA RECOVERED =    5112
AREA OF  BOARD =    8064
PRCNT RECOVERY = 63.3929

BOARD NBR =  303
COMBO NBR =    7
   0   5  14  15  16  17
AREA RECOVERED =    5884
AREA OF  BOARD =    8064
PRCNT RECOVERY = 72.9663

BOARD NBR =  304
COMBO NBR =    7
   0   5  14  15  16  17
AREA RECOVERED =    5908
AREA OF  BOARD =    8064
PRCNT RECOVERY = 73.2639

BOARD NBR =  305
COMBO NBR =    7
   0   5  14  15  16  17
AREA RECOVERED =    6460
AREA OF  BOARD =    8064
PRCNT RECOVERY = 80.1091


TOTAL AREA RECOVERED =     71028
TOTAL AREA OF BOARDS =     94080
PRCNT TOTAL RECOVERY =   75.4974
```

```
TOTAL AREA RECOVERED-3C =    41804
TOTAL AREA OF BOARDS-3C =    53760
PCNT RECOVERY BOARDS-3C =  77.7604


TOTAL AREA RECOVERED-4C =    29224
TOTAL AREA OF BOARDS-4C =    40320
PCNT RECOVERY BOARDS-4C =  72.4802


PRCNT RECOVERY PER CUTTING SIZE   BOARDS-3C
 26.68899    51.07143    0.00000    0.00000

PRCNT RECOVERY PER CUTTING SIZE   BOARDS-4C
 24.88095    47.59921    0.00000    0.00000

COMBO NBR    NBR TIMES USED
    5              5
    7              5

END OF RUN. BOARDS  291 -   305 PROCESSED
```

An example of the System and Program Control Cards for running program CONST/RECVRY which generated the sample output follows:

```
JLIMITS, CM55000, T30, PR30.  ACOUNT NBR., NAME.
FTN.
REQUEST, TAPE7, D0000, 01.    CONST OUPUT (DISK)
LGO.
REWIND (TAPE7,LGO)              RECVRY INPUT (DISK)
RFL(43000)
FTN.
LGO.
7/8/9
CONST Source Deck.
7/8/9
CONST Program Control Card (LBTD,NBSTR,NBEND).
Board Defect Data Cards (NDEF+3 Cards/Board).
7/8/9
RECVRY Source Deck.
7/8/9
RECVRY Program Control Card Type 1 (TITLE).
RECVRY Program Control Card Type 2 (LBTD, LPRM,
        NBSTR, NBEND).
6/7/8/9 END-OF-JOB Card.
```

If Programs CONST and RECVRY are to be run as two separate programs, request TAPE7 to be a magnetic tape output file from CONST and specify it as a tape input file to Program RECVRY.

Appendix 2: Source Listings for Programs

Program PERMU

```
      PROGRAM PERMU
     1(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE1)
C
C CONVERSION OF PROG NAME J1 (BASIC) TO FORTRAN. THIS PROGRAM CONSTRUCTS
C THE PERMUTATION, COMBINATION TABLE FOR NSIZE(NBR OF BRD WDTH SIZES)
C AND NSWBL(NBR OF MACHINE SAW BLADES). INITIAL INPUT IS TAKEN FROM TWO
C CARDS. OUTPUT IS A PRINT LISTING AND IF DESIRED A TAPE FILE(TAPE1).
C D. MARKSTROM  10/72  RMFRES  BJE
C
C--INPUT CARD 1  (3I5)
C      IBWDT=BOARD WIDTH  14,30,46 UNITS
C      NSIZE=NBR OF DIFFERENT SIZES TO CONSIDER. MAX=5
C      NSWBL=NBR OF SAW BLADES FOR YOUR MACHINE. MAX=6
C--INPUT CARD 2  (5I5)
C      KUT(I)=SIZES FOR NSIZE SIZES ON CARD 1.   MAX=5
C
C--PROGRAM VARIABLE NAMES
C      P(I,J)=PERMUTATION ARRAY FOR 6 SAW BLADES, 5 CUTTING WIDTHS
C      C(I)=COMBINATION ARRAY FOR 6 SAW BLADES, 5 CUTTING WIDTHS
C      LSUM=TOTAL NBR OF DIFFERENT CUT COMBINATIONS IN ARRAY C(KK)
C      NSWB=SAW BLADE NBR., 1 TO NSWBL+1
C      NCMB=COMBINATION (COMBO) NBR., 1 TO 1023. (FOR 3 SIZES,5 BLADES)
C      KERF=ACCUMULATOR FOR 1/4-INCH SAW KERFS
C      MWDH=ACCUMULATOR FOR CUTTING WIDTHS IN 1/4-INCH UNITS
C
      INTEGER P(7,1023),C(10)
      DIMENSION KUT(10)
      REWIND 1
C
C--INITIALIZE ARRAYS AND VARIABLES
      DO 5 I=1,10
      C(I)=0
      KUT(I)=0
    5 CONTINUE
      DO 8 N1=1,7
      DO 7 N2=1,1023
      P(N1,N2)=0
    7 CONTINUE
    8 CONTINUE
      LSUM=0
C
C--READ AND PRINT OUT INPUT CARDS 1 AND 2. WRITE TAPE FILE 1 ID RECORD
      READ(5,100) IBWDT,NSIZE,NSWBL
  100 FORMAT(5I5)
      KK=NSWBL + 1
      DO 15 N1=1,KK
      DO 10 N2=1,1023
      P(N1,N2)=6
   10 CONTINUE
   15 CONTINUE
      READ(5,100) (KUT(I),I=1,5)
      WRITE(6,102) IBWDT,NSIZE,NSWBL,(KUT(I),I=1,5)
  102 FORMAT(1H1,3I5,5X,5I5,/)
      WRITE(1,103) NSIZE,(KUT(I),I=1,5)
  103 FORMAT(6I5)
C
C--DETERMINE COMBINATIONS FOR NSIZE CUTS AND KK SAW BLADES
      K=0
      C(2)=NSIZE
      LSUM=LSUM + NSIZE
      I=0
      DO 20 K=2,KK
      I=K + 1
      C(I)=NSIZE**K + C(I-1)
      LSUM=LSUM + NSIZE**K
   20 CONTINUE
C
      NCUT=C(I-1)
      WRITE(6,104) NCUT
  104 FORMAT(1H ,*NBR OF DIFFERENT CUT COMBINATIONS =*I6,/)

C
C--FILL ARRAY P FOR NSIZE CUTS AND KK SAW BLADES
      N=1
      I=0
      C(1)=1
      DO 50 NSWB=1,KK
      K=C(NSWB)
      I=NSWB - 1
      IF(I.EQ.0) KZ=1
      IF(I.EQ.0) GO TO 25
      KZ=NSIZE**I
      K=K + 1
   25 KCZ=K + KZ
      DO 30 NCMB=K,KCZ
      IF(NCMB.GT.1023) GO TO 35
      P(NSWB,NCMB)=N
   30 CONTINUE
   35 N=N + 1
      K=K + KZ
      IF(N.GT.NSIZE) GO TO 40
      IF(K.GE.LSUM) GO TO 45
      GO TO 25
   40 N=1
      GO TO 25
   45 N=1
   50 CONTINUE
C
C--COMBINE CUT WIDTH AND SAW KERF SELECTIONS AND DETERMINE MIN BOARD
C--WIDTH FROM WHICH CUT CAN BE MADE
      KERF=0
      MWDH=0
      DO 75 NCMB=1,1023
      DO 55 NSWB=1,KK
      L=P(NSWB,NCMB)
      IF(L.EQ.6) GO TO 60
      MWDH=MWDH + KUT(L)
      KERF=KERF + 1
   55 CONTINUE
      GO TO 70
   60 MSUM=MWDH + KERF

      IF(MSUM.LE.IBWDT) GO TO 65
      P(7,NCMB)=99
      GO TO 70
   65 P(7,NCMB)=MWDH
   70 KERF=0
      MWDH=0
   75 CONTINUE
C
C--PRINT OUT PERM-COMB TABLE, ARRAY P, OF POSSIBLE CUTS FROM A BOARD
C--4 UNITS(1-INCH) TO IBWDT UNITS WIDE.
      KTMP=0
      DO 85 N=4,IBWDT
      DO 80 NCMB=1,1023
      IF(P(7,NCMB).NE.N) GO TO 80
      MWDH=P(7,NCMB)
      KTMP=P(6,NCMB)
      IF(KTMP.NE.0) GO TO 78
      KTMP=6
   78 WRITE(6,106) NCMB,(P(I,NCMB),I=1,5),KTMP,MWDH
  106 FORMAT(1H ,8I5)
      WRITE(1,108) NCMB,(P(I,NCMB),I=1,5),KTMP,MWDH
  108 FORMAT(8I5)
   80 CONTINUE
   85 CONTINUE
C
      ENDFILE 1
      REWIND 1
      CALL EXIT
      END
```

Program CONST

```
      PROGRAM CONST
     1(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE7)
C
C CONVERSION OF PROG NAME CONST (BASIC) TO FORTRAN. THIS PROGRAM INPUTS
C INITIAL X,Y DEFECT COORDINATES(FROM  CARDS) AND OUTPUTS MODIFIED X,Y
C COORDINATES(ON FILE 7) USING A MIN LENGTH BETWEEN DEFECTS LIMITATION
C WHICH IS CARD INPUT(VAR. LBTD). THE MODIFIED X,Y FOR DEFECTS IS INPUT
C TO PROG RECVRY WHICH SELECTS THE CUT COMBO FOR EACH INDIVIDUAL BRD.
C      D. MARKSTROM  10/72  RMFRES  BJE  MODF. 07/72
C
C--INPUT CONTROL CARD (3I5)
C      LBTD=MIN. LENGTH BETWEEN DEFECTS CONTROL VARIABLE (IN UNITS)
C      NBSTR=NBR OF THE BRD TO START ANALYSIS  EXP. 201
C      NBEND=NBR OF THE BRD TO END ANALYSIS    EXP. 299
C
C--DATA CARDS. BRD IDENT., BRD COORD., DEF COORD., END OF BRD INPUT.
C
C--BOARD IDENTIFICATION CARD. ONE/BOARD (5X,I2,24X,I5,32X,I4)
C      NGRD=BOARD GRADE CODE
C      NBRD=BOARD NUMBER
C      NDEF=NUMBER OF DEFECTS IN BOARD
C--BOARD COORDINATE CARD. ONE/BOARD (I3,1X,I3,5X,I3,1X,I3)
C      IBXS=BOARD X-DIR(WIDTH) START COORD. IBXS=0
C      IBYS=BOARD Y-DIR(LNGTH) START COORD. IBYS=0
C      IBXE=BOARD X-DIR(WIDTH) END   COORD. IBXE=14,30, OR 46
C      IBYE=BOARD Y-DIR(LNGTH) END   COORD. IBYE=576,672, OR 768
C--DEFECT COORDINATE CARD(S). NDEF/BOARD (I3,1X,I3,5X,I3,1X,I3)
C      IDXS=DEFECT X-DIR(WIDTH) START COORD. IDXS GE IBXS
C      IDYS=DEFECT Y-DIR(LNGTH) START COORD. IDYS GE IBYS
C      IDXE=DEFECT X-DIR(WIDTH) END   COORD. IDXE LE IBXE
C      IDYE=DEFECT Y-DIR(LNGTH) END   COORD. IDYE LE IBYE
C--END OF BOARD DATA CARD. ONE/BOARD (I3,1X,I3,5X,I3,1X,I3)
C      IDXS=999
C      IDYS=0
C      IDXE=0
C      IDYE=0
C--A SET OF DATA CARDS FOR ONE BOARD WILL CONTAIN NDEF+3 CARDS.
C
C--PROGRAM VARIABLE NAMES
C      T(I,44)=TABLE OF START AND END DEFECT COORD. FOR A BRD. I=IBXE
C      DFCT(3)=ACCUMULATOR FOR DEFECT AREA FOR 3 CUTTINGS, 4-8-12 UNITS.
C      P(I)=POINTER WHICH POSITIONS DEFECT COORD. IN TABLE T.  I=IBXE
C      KUT(3)=THREE CUTTING SIZES.  SET BY PROG STATEMENTS.
C      LWB(I,768)=BRD DIMENSION ARRAY FOR COMPUTER SIMULATION OF ALL
C                  DEFECTS IN A BRD. I=IBXE
C      MIND(I)=MIN DEFECT COORD. FOR LWB SCAN OF DEFECT AREA.  I=IBXE
C      MAXD(I)=MAX DEFECT COORD. FOR LWB SCAN OF DEFECT AREA.  I=IBXE
C      LDS=LENGTH DEFECT START COORD. VALUE, USED TO X-OUT DEFECT IN LWB
C      LDE=LENGTH DEFECT END   COORD. VALUE, USED TO X-OUT DEFECT IN LWB
C      NX=WIDTH DEFECT COORD. VALUE,         USED TO X-OUT DEFECT IN LWB
C
C--THE I DIMENSION IS SET BY THE USER AND IS DETERMINED BY THE WIDTH
C--SIZE OF BOARDS TO BE PROCESSED. THIS HELPS TO REDUCE CORE STORAGE
C--REQUIREMENTS AND RUN COSTS.
C
      INTEGER T(14,44),DFCT(3),P(14)
      DIMENSION KUT(3)
      COMMON LWB(14,768),MIND(14),MAXD(14),LDS,LDE,NBRD,NX
      REWIND 7
C--INITIALIZE ARRAYS. SET THE UPPER LIMIT OF THE I INDEX TO IBXE.
      KFRST=0
      KOUNT=0
      KUT(1)=3
      KUT(2)=7
      KUT(3)=11
      DFCT(1)=0
      DFCT(2)=0
      DFCT(3)=0
      DO 2 I=1,14
      DO 2 J=1,768
    2 LWB(I,J)=0
      DO 4 I=1,14
      DO 4 J=1,44
    4 T(I,J)=0
      DO 6 I=1,14
    6 P(I)=2
C--READ INPUT CONTROL CARD
      READ(5,100) LBTD,NBSTR,NBEND
  100 FORMAT(3I5)
      WRITE(6,101) LBTD,NBSTR,NBEND
  101 FORMAT(1H1,5X,*MIN. LENGTH BETWEEN DEFECTS CONTROL FOR THIS RUN IS
     1 *I4,2X,I4,* - *,I4,/)
C
   10 DO 8 I=1,14
      MIND(I)=900
    8 MAXD(I)=0
C--READ BOARD ID CARD
   12 READ(5,102) NGRD,NBRD,NDEF
  102 FORMAT(5X,I2,24X,I5,32X,I4)
      IF(EOF(5)) 90,15
   15 IF(NBRD.LT.NBSTR) GO TO 87
      IF(NBRD.GT.NBEND) GO TO 90
      WRITE(6,104) NGRD,NBRD,NDEF
  104 FORMAT(1H0,I3,1X,I3,5X,I3)
C--READ BOARD COORD. CARD
      READ(5,106) IBXS,IBYS,IBXE,IBYE
  106 FORMAT(I3,1X,I3,5X,I3,1X,I3)
C--WRITE FILE 7 FOR RECVRY INPUT
      WRITE(7) NGRD,NBRD,NDEF
      WRITE(7) IBXS,IBYS,IBXE,IBYE
      WRITE(6,107) IBXS,IBYS,IBXE,IBYE
  107 FORMAT(1H ,I3,1X,I3,5X,I3,1X,I3)
C--SET BRD Y-DIR START COORD. = 1  IBYS
      IBYS=1
C--READ BOARD DEFECT COORD. CARD(S)
   20 READ(5,106) IDXS,IDYS,IDXE,IDYE
      IF(IDXS.EQ.999) GO TO 35
C     IF(KFRST.NE.0) GO TO 22
      WRITE(6,107) IDXS,IDYS,IDXE,IDYE
   22 LDS=IDYS + 1
      LDE=IDYE
      IF(IDXE.LE.IBXE) GO TO 25
      IDXE=IBXE
   25 KK=IDXE - 1
      IF(IDXS.EQ.0.AND.KK.EQ.0) GO TO 32
C--STORE DEFECT COORD. IN ARRAY T AND SIMULATE DEFECT IN ARRAY LWB
      DO 30 I=IDXS,KK
      N=I + 1
      K=P(N)
      T(N,K+1)=IDYS
      T(N,K+2)=IDYE
      P(N)=P(N) + 2
      NX=N
      CALL XOUT
   30 CONTINUE
      GO TO 20
C
   32 K=P(1)
      T(1,K+1)=IDYS
      T(1,K+2)=IDYE
      P(1)=P(1) + 2
      NX=1
      CALL XOUT
      GO TO 20
C
C--DETERMINE IF MIN LENGTH BETWEEN DEFECTS EXISTS. IF NO, BLOCK OUT
C--(X-OUT) CLEAR AREA BETWEEN DEFECTS IN ARRAY LWB.
   35 CONTINUE
      KEND=IBXE
      DO 50 N=1,KEND
      K=P(N)
      P(N)=P(N) + 1
      T(N,K+1)=IBYE
      T(N,K+2)=IBYE
      KK=P(N)
      DO 45 M=3,KK,2
      IF(T(N,M) - T(N,M-1).EQ.0) GO TO 45
      IF(T(N,M) - T(N,M-1).LT.LBTD) GO TO 40
      IF(T(N,M).EQ.IBYE) GO TO 45
      GO TO 45
   40 LDS=T(N,M-1) + 1
      LDE=T(N,M+1)
      NX=N
      CALL XOUT
   45 CONTINUE
      P(N)=2
   50 CONTINUE
C
C--SET INDEXS FOR BOARD SCAN TO FIND DEFECT AREAS
      DO 85 N=1,KEND
      KK=0
      IS=N
      DO 80 K=3,5
      I=K - KK
      KK=KK + 2
      IE=N + KUT(I)
      LIMIT=IBXE - KUT(I)
      IF(N.GT.LIMIT) GO TO 80
C
C--FIND MIN-MAX DEFECT COORD. FOR BOARD SCAN. IF MAXD(M)=0, NO DEFECTS
C--FOR RANGE OF SCAN.
C
      DO 54 M=IS,IE
      IF(MAXD(M).EQ.0) GO TO 54
      GO TO 56
   54 CONTINUE
      GO TO 80
   56 IBYS=MIND(IS)
      IBYE=MAXD(IS)
      DO 60 M=IS,IE
      IF(MIND(M).LT.IBYS) IBYS=MIND(M)
      IF(MAXD(M).GT.IBYE) IBYE=MAXD(M)
   60 CONTINUE
C
C--MAKE X,Y SCAN OF BOARD ARRAY LWB TO FIND DEFECT AREA. IF DEFECT FOUND
C--ACCUMULATE DEFECT AREA IN DFCT(I).
      NCLRY=0
      DO 75 J=IBYS,IBYE
      DO 65 M=IS,IE
      IF(LWB(M,J).EQ.NBRD) GO TO 68
   65 CONTINUE
      NCLRY=NCLRY + 1
      GO TO 75
C--TEST FOR MIN DISTANCE BETWEEN DEFECTS IN LENGTH(Y) DIRECTION
   68 IF(NCLRY.EQ.0) GO TO 70
      IF(NCLRY.GE.LBTD) NCLRY=0
      IF(NCLRY.GE.LBTD) GO TO 70
      MULT=KUT(I) + 1
      IADD=NCLRY * MULT
      DFCT(I)=DFCT(I) + IADD
      NCLRY=0
C
   70 DFCT(I)=DFCT(I) + KUT(I) + 1
   75 CONTINUE
   80 CONTINUE
      P(1)=2
C
```

```
C--LIST DEFECT AREAS FDR FIRST BOARD ANO WRITE FILE 7 FOR ALL BOARDS
C--FOR RECVRY INPUT.  BRANCH AND DO NEXT BOARO.
C      IF(KFRST.NE.0) GD TO 82
       WRITE(6,108) N,DFCT(1),DFCT(2),DFCT(3)
  108 FDRMAT(1H ,I4,3(2X,I5))
   82 WRITE(7) DFCT(1),DFCT(2),OFCT(3)
       DFCT(1)=0
       DFCT(2)=0
       OFCT(3)=0
   B5 CONTINUE
       KDUNT=KDUNT + 1
       KFRST=1
       GD TO 10
C
   87 KRECS=NDEF + 2
       DD BB NSKP=1,KRECS
       READ(5,106) IDUM1,IDUM2,IDUM3,IDUM4
   BB CDNTINUE
       GO TD 12
C
   90 WRITE(6,110) KOUNT
  110 FORMAT(1H0,5X,*END-OF-RUN. TOTAL BOARDS PROCESSED =*I4)
       ENOFILE 7
       REWINO 7
       ENO

       SUBRDUTINE XDUT
C
C RDUTINE GD SUB 490 FROM PROG CONST (BASIC)
C THIS ROUTINE SIMULATES THE BRO DEFECT AREAS BY STORING THE BRO NUMBER
C (NBRO) IN ALL X,Y LDCATIDNS OF A DEFECT. IT IS X-ING DUT, OR BLOCKING
C DUT THE AREA OF DEFECT AS SPECIFIED BY THE INOICATORS NX,LOS, ANO LDE.
C MAX AND MIN DEFECT LIMITS IN THE Y DIRECTION ARE ALSO DETERMINED.
C
       COMMDN LWB(14,768),MIND(14),MAXD(14),LDS,LDE,NBRD,NX
       OD 10 I=LDS,LDE
       LWB(NX,I)=NBRD
   10 CONTINUE
C
       IF(LDE.GT.MAXD(NX)) MAXD(NX)=LDE
       IF(LOS.LT.MINO(NX)) MIND(NX)=LDS
       RETURN
       ENO
```

Program RECVRY

```
      PROGRAM RECVRY
     1(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE7)
C
C CONVERSION OF PROG NAME G0001 (BASIC) TO FORTRAN. THIS PROGRAM INPUTS
C THE MODIFIED X,Y BOARO COORDINATES (DEFECT AREA) FROM FILE TAPE7
C GENERATEO IN PROG CONST AND OUTPUTS THE BEST CUT TO MAXIMIZE RECOVER-
C ABLE AREA FOR INOIVIOUAL BOARDS. THE PERMU-COMBO TABLE TO BE USEO TO
C EVALUATE SELECTEO BRO SIZES IS CODED IN DATA ARRAY KPCT. THESE VALUES
C WILL CHANGE OEPENOING ON BRD SIZE AND A NEW OATA LIST MUST BE INSERTED
C IN SOURCE DECK.  D. MARKSTROM  1D/72  RMFRES  BJE  MODF. D6/72
C
C--INPUT CONTROL CARO TYPE 1 (4A10)
C     TITLE(I)=IDENTIFICATION CARD FOR RUN
C
C--INPUT CONTROL CARD TYPE 2 (4I5)
C     LBTD=MIN. LENGTH BETWEEN OEFECTS, SAME AS IN PROG CONST.
C     LPRM=LOWER LIMIT OF COMBINED CUTTINGS TO BE CONSIDEREO FROM THE
C          PERMU-COMBO SETS IN ARRAY KPCT. CUTTINGS GT LPRM ARE USED
C          TO EVALUATE BEST FIT UNTILL A STOP INOICATOR, 999 IS FOUNO.
C     NBSTR=NBR OF THE BRO TO START ANALYSIS  EXP. 201
C     NBEND=NBR OF THE BRO TO END ANALYSIS    EXP. 299
C
C--PROGRAM VARIABLE NAMES.
C
C     NSIZE=NBR OF CUTTING SIZES TO CONSIDER  MAX OF 5
C     KUT(I)=CUT SIZES IN QTR UNITS FOR NSIZE CUTS  EXP. 4,8,12,D,D
C     NGRD=BRO GRADE COOE
C     NBRO=BRO NUMBER
C     NDEF=NBR OF OEFECTS IN BRD
C     IBXS=BRO X-DIR(WIOTH) START COORO. IBXS=0
C     IBYS=BRD Y-OIR(LNGTH) START COORD. IBYS=D
C     IBXE=BRO X-OIR(WIDTH) ENO   COORO. IBXE=14,3D,OR 46
C     IBYE=BRO Y-OIR(LNGTH) ENO   COORO. IBYE=576,672,OR 768
C     NCMB=COMBINATION(COMBO) NUMBER
C     KKS1=COMBO CUT SIZE 1    COOE 1=4, 2=B, 3=12 UNITS
C     KKS2=COMBO CUT SIZE 2    COOE 1=4, 2=8, 3=12 UNITS
C     KKS3=COMBO CUT SIZE 3    CODE 1=4, 2=B, 3=12 UNITS
C     KKS4=COMBO CUT SIZE 4    COOE 1=4, 2=8, 3=12 UNITS
C     KKS5=COMBO CUT SIZE 5    COOE 1=4, 2=8, 3=12 UNITS
C     KKS6=COMBO CUT SIZE 6    COOE 1=4, 2=B, 3=12 UNITS
C     NTCUT=TOTAL UNITS CUT FOR COMBO NCMB(SAWKERFS NOT INCLUOEO)
C
C     CLEAR(I,J)=WORK ARRAY TO OETERMINE CLEAR AREA OF BOARO
C     RGRSZ(I,J)=ARRAY TO ACCUMULATE RECOVERY PER BOARO GRAOE ANO SIZE
C     SCMBN(I)=ARRAY OF SELECTEO COMBO NBRS
C     IGRO(I)=TOTAL AREA RECOVEREO BY GRAOE
C     JGRD(I)=TOTAL AREA OF BOAROS BY GRAOE
C
      INTEGER CLEAR(6,5D),RGRSZ(5,6),SCMBN(13D),TITLE(4),P1,P2,P3,P4,P5,
     1P6,CN
      OIMENSION KUT(9),IGRO(9),JGRO(9),KSZE(6)
      OIMENSION KPCT(56)
C
C--OATA LIST STATEMENTS. KPCT IS OEPENOENT ON SIZE OF BROS EVALUATEO.
C--INCLUDE ONE MORE COMBO SET THEN OESIREO TO TERMINATE SEARCH WITH
C--NCMB (KPCT(IDX)) SET = 999. KPCT IS FOR IBXE=14 BOARO WIOTH.
C
      OATA KSZE/3,4,8,12,D,0/
      OATA KPCT/1,1,6,6,6,6,6,4,2,2,6,6,6,6,6,B,4,1,1,6,6,6,6,8,3,3,6,6,
     16,6,6,12,5,2,1,6,6,6,6,12,7,1,2,6,6,6,6,12,999,1,1,1,1,1,1,99/
C
C--ZERO OUT ARRAYS ANO REAO CONTROL CAROS 1 ANO 2
      KA=0
      KSAR=0
      KSAB=0
      KSLT=1
      REWINO 7
      OO 2 I=1,6
      OO 2 J=1,50
    2 CLEAR(I,J)=0
      OO 4 I=1,5
      OO 4 J=1,6
    4 RGRSZ(I,J)=0
      OO 6 I=1,130
    6 SCMBN(I)=0
      DO 10 I=1,9
      KUT(I)=D
      IGRO(I)=0
   1D JGRO(I)=0
      REAO(5,1DO) TITLE
  10D FORMAT(4A1D)
      REAO(5,1D2) LBTO,LPRM,NBSTR,NBENO
  102 FORMAT(4I5)
      WRITE(6,104) TITLE,LBTO,LPRM,NBSTR,NBENO
  104 FORMAT(1H1,9X,4A1D,4I5,//)
      NSIZE=KSZE(1)
      OO 15 I=1,5
   15 KUT(I)=KSZE(I+1)
C
C--REAO INPUT GENERATEO BY PROG CONST. IO, BRO COORO., OEFECT AREAS
   2D IOX=1
      REAO(7) NGRO,NBRO,NOEF
      IF(EOF(7)) 90,25
   25 READ(7) IBXS,IBYS,IBXE,IBYE
      IF(NBRO.LT.NBSTR) GO TO 27
      IF(NBRO.GT.NBENO) GO TO 90
      GO TO 28
   27 KSLT=D
C
   2B M1=IBXE
      OO 3D J=1,M1
      REAO(7) CLEAR(1,J),CLEAR(2,J),CLEAR(3,J)
   3D CONTINUE
      IF(KSLT.EQ.0) GO TO 80
C
C--USING DEFECT AREAS FROM PROG CONST, COMPUTE CLEAR AREAS OF BOARD
      OO 40 I=1,NSIZE
      MULT=KUT(I)*IBYE
      M1=((IBXE-KUT(I))+1)
      OO 35 J=1,M1
      CLEAR(I,J)=MULT - CLEAR(I,J)
   35 CONTINUE
   40 CONTINUE
      GO TO 50
C
C     OO 45 J=1,IBXE
C     WRITE(6,112) J,CLEAR(1,J),CLEAR(2,J),CLEAR(3,J)
C 112 FORMAT(1H ,4I8)
C  45 CONTINUE
C
C--SELECT A CUTTING COMBINATION FROM ARRAY KPCT
   5D CONTINUE
   52 NCMB=KPCT(IDX)
      KKS1=KPCT(IOX+1)
      KKS2=KPCT(IDX+2)
      KKS3=KPCT(IDX+3)
      KKS4=KPCT(IDX+4)
      KKS5=KPCT(IOX+5)
      KKS6=KPCT(IDX+6)
      NTCUT=KPCT(IDX+7)
      IF(NTCUT.EQ.99) GO TO 75
      IOX=IOX+8
      IF(NTCUT.LE.LPRM) GO TO 52
      KX=IBXE - NTCUT
C
C--KX IS NBR OF SAW KERF CUTS WITHIN THE BOARO OIMENSIONS IBXS-IBXE
C--EVALUATE SELECTEO CUTTING COMBINATION FOR MAX CLEAR AREA, STORE IN KA
      OO 7D N=1,KX
      I1=(KUT(KKS1) + N + 1)
      J1=I1 + (KX-N)
      KS=CLEAR(KKS1,N)
      DO 6B N1=I1,J1
      KS1=CLEAR(KKS2,N1) + KS
      I2=(KUT(KKS2) + N1 + 1)
      J2=I2 + (KX-((J1-I1)+N))
      OO 66 N2=I2,J2
      KS2=CLEAR(KKS3,N2) + KS1
      I3=(KUT(KKS3) + N2 + 1)
      J3=I3 + (KX-((J2-I2)+(J1-I1)+N))
      OO 64 N3=I3,J3
      KS3=CLEAR(KKS4,N3) + KS2
      I4=(KUT(KKS4) + N3 + 1)
      J4=I4 + (KX-((J3-I3)+(J2-I2)+(J1-I1)+N))
      OO 62 N4=I4,J4
      KS4=CLEAR(KKS5,N4) + KS3
      I5=(KUT(KKS5) + N4 + 1)
      J5=I5 + (KX-((J4-I4)+(J3-I3)+(J2-I2)+(J1-I1)+N))
      OO 6D N5=I5,J5
      KS5=CLEAR(KKS6,N5) + KS4
      IF(KS5.GT.KA) GO TO B5
   58 KS5=D
   6D CONTINUE
   62 CONTINUE
   64 CONTINUE
   66 CONTINUE
   68 CONTINUE
   70 CONTINUE
      GO TO 52

C
C--OUTPUT INOIVIOUAL BOARO RESULTS, PCNT RECOVERY BY BRO
   75 IF(KA.EQ.D) GO TO BD
      WRITE(6,116) NBRD
  116 FORMAT(1H ,9X,*BOARO NBR = *I4)
      WRITE(6,118) CN
  118 FORMAT(1H ,9X,*COMBO NBR = *I4)
      WRITE(6,12D) P1,P2,P3,P4,P5,P6
  120 FORMAT(1H ,9X,6I4)
      WRITE(6,122) KA
  122 FORMAT(1H ,9X,*AREA RECOVEREO =  *I6)
      KB=IBXE*IBYE
      WRITE(6,124) KB
  124 FORMAT(1H ,9X,*AREA OF  BOARO =  *I6)
      AR=FLOAT(KA)
      AB=FLOAT(KB)
      PR=(AR/AB) * 1DD.
      WRITE(6,126) PR
  126 FORMAT(1H ,9X,*PRCNT RECOVERY =*F8.4,/)
      P1=P1 + 1
      P2=P2 + 1
      P3=P3 + 1
      P4=P4 + 1
      P5=P5 + 1
      P6=P6 + 1
C
C--ACCUMULATE RESULTS BY GRAOE ANO CUTTING SIZES
      SCMBN(CN)=SCMBN(CN) + 1
      KSAB=KSAB + KB
      KSAR=KSAR + KA
      IGRO(NGRO)=IGRO(NGRO) + KA
      JGRO(NGRO)=JGRD(NGRO) + KB
      RGRSZ(NGRO,L1)=RGRSZ(NGRO,L1) + CLEAR(L1,P1)
      RGRSZ(NGRD,L2)=RGRSZ(NGRO,L2) + CLEAR(L2,P2)
      RGRSZ(NGRO,L3)=RGRSZ(NGRO,L3) + CLEAR(L3,P3)
      RGRSZ(NGRO,L4)=RGRSZ(NGRO,L4) + CLEAR(L4,P4)
      RGRSZ(NGRO,L5)=RGRSZ(NGRO,L5) + CLEAR(L5,P5)
      RGRSZ(NGRO,L6)=RGRSZ(NGRD,L6) + CLEAR(L6,P6)
```

```
C
   B0 KA=0
      KSLT=1
      00 B2 I=1,6
      00 B2 J=1,50
   B2 CLEAR(I,J)=0
      GO TO 20
C
   B5 CN=NCMB
      KA=KS5

      P1=N - 1
      P2=N1 - 1
      P3=N2 - 1
      P4=N3 - 1
      P5=N4 - 1
      P6=N5 - 1
      L1=KKS1
      L2=KKS2
      L3=KKS3
      L4=KKS4
      L5=KKS5
      L6=KKS6
      GO TO 5B
C
C--OUTPUT ALL BOARO RESULTS, PCNT RECOVERY BY GRAOE, BY SIZE WITHIN GRO
   90 WRITE(6,200) KSAR
  200 FORMAT(1H0,9X,*TOTAL AREA RECOVEREO = *IB)
      WRITE(6,202) KSAB
  202 FDRMAT(1H ,9X,*TOTAL AREA OF BOARDS = *IB)
      AR=FLOAT(KSAR)
      AB=FLDAT(KSAB)
      PR=(AR/AB) * 100.
      WRITE(6,204) PR
  204 FORMAT(1H ,9X,*PRCNT TOTAL RECOVERY = *FB.4,/)
C
      WRITE(6,206) IGRO(3)
  206 FORMAT(1H0,9X,*TOTAL AREA RECOVEREO-3C = *IB)
      WRITE(6,20B) JGRO(3)
  20B FORMAT(1H ,9X,*TOTAL AREA OF BOAROS-3C = *IB)
      AR=FLOAT(JGRO(3))
      AB=FLOAT(IGRO(3))
      IF(AR.EQ.0.0) PR=0.0
      IF(AR.EQ.0.0) GO TO 92
      PR=(AB/AR) * 100.
   92 WRITE(6,210) PR
  210 FORMAT(1H ,9X,*PCNT RECOVERY BOAROS-3C = *FB.4,/)
C
      WRITE(6,212) IGRD(4)
  212 FORMAT(1H0,9X,*TOTAL AREA RECOVEREO-4C = *IB)
      WRITE(6,214) JGRO(4)
  214 FORMAT(1H ,9X,*TOTAL AREA OF BOAROS-4C = *IB)
      AR=FLOAT(JGRD(4))
      AB=FLOAT(IGRO(4))
      IF(AR.EQ.0.0) PR=0.0
      IF(AR.EQ.0.0) GO TO 94
      PR=(AB/AR) * 100.
   94 WRITE(6,216) PR
  216 FORMAT(1H ,9X,*PCNT RECOVERY BOAROS-4C = *FB.4,/)
C
      WRITE(6,21B)
  21B FORMAT(1H0,9X,*PRCNT RECOVERY PER CUTTING SIZE  BOAROS-3C*)
      IF(JGRO(3).EQ.0) GO TO 96
      AR=FLOAT(JGRD(3))
      AB=FLOAT(RGRSZ(3,1))

      PR1=(AB/AR) * 100.
      AB=FLOAT(RGRSZ(3,2))
      PR2=(AB/AR) * 100.
      AB=FLDAT(RGRSZ(3,3))
      PR3=(AB/AR) * 100.
      AB=FLOAT(RGRSZ(3,4))
      PR4=(AB/AR) * 100.
      GO TO 9B
   96 PR1=0.0
      PR2=0.0
      PR3=0.0
      PR4=0.0
   9B WRITE(6,220) PR1,PR2,PR3,PR4
  220 FORMAT(1H ,7X,4(2X,F9.5))
C
      WRITE(6,222)
  222 FORMAT(1H0,9X,*PRCNT RECOVERY PER CUTTING SIZE  BOARDS-4C*)
      IF(JGRO(4).EQ.0) GO TO 140
      AR=FLOAT(JGRO(4))
      AB=FLDAT(RGRSZ(4,1))
      PR1=(AB/AR) * 100.
      AB=FLOAT(RGRSZ(4,2))
      PR2=(AB/AR) * 100.
      AB=FLDAT(RGRSZ(4,3))
      PR3=(AB/AR) * 100.
      AB=FLOAT(RGRSZ(4,4))
      PR4=(AB/AR) * 100.
      GO TO 142
  140 PR1=0.0
      PR2=0.0
      PR3=0.0
      PR4=0.0
  142 WRITE(6,220) PR1,PR2,PR3,PR4
C
      WRITE(6,224)
  224 FORMAT(1H0,9X,*COMBO NBR   NBR TIMES USEO *)
      DO 144 KC=1,130
      IF(SCMBN(KC).EQ.0) GO TO 144
      WRITE(6,226) KC,SCMBN(KC)
  226 FORMAT(1H ,9X,I5,10X,I5)
  144 CONTINUE
C
C--ENO DF RUN. REWIND TAPE7, PRINT MESSAGE, CALL EXIT
C
      WRITE(6,22B) NBSTR,NBEND
  22B FORMAT(1H0,9X,*END DF RUN. BOAROS*I5,* - *I5,* PROCESSEO*)
      REWINO 7
      CALL EXIT
      ENO
```

Program DEFEDIT

```
      PROGRAM DEFEDIT(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
C
C PROG TO EDIT DATA CARDS FOR CUTUP-REPAIR-RECOVERY STUDY  D.MARKSTROM
C
      DIMENSION DLW(30),DLL(30),DUW(30),DUL(30),IDH(B)
C
      READ(5,200) IDH
  200 FORMAT(8A10)
      WRITE(6,1) IDH
    1 FORMAT(1H1,5X,8A10,/)
C--READ AND CHECK HEADER CARD.
    3 READ(5,10) CLASS, NUMBD,NDEF
   10 FORMAT(5X,R3,22X,I6,32X,I4)
      IF(EOF(5)) 100,9
    9 IF(CLASS .EQ. 3R 3C .OR. CLASS .EQ. 3R 4C)   GOTO 11
      GO TO 14
   11 IF(NDEF .LT. 1) GO TO 14
      GO TO 13
   14 WRITE(6,2) NUMBD
    2 FORMAT(*0*,10X,*ERROR IN THE FIRST CARD OF BOARD *,I4)
C--READ BOARD COORDINATE CARD. CHECK LWR COORD. LT UPR COORD..
   13 READ(5,20) PLW,PLL,PUW,PUL
   20 FORMAT(F3.0,1X,F3.0,5X,F3.0,1X,F3.0)
      IF(PLW .GE. PUW .OR. PLL .GE. PUL) GO TO 46
      IF(PUW .NE. 14.)GO TO 21
      GO TO 25
   21 IF(PUW .NE. 30.)GO TO 22
      GO TO 25
   22 IF(PUW .EQ. 46.)GO TO 25
      GO TO 46
   25 IF(PUL .NE. 576.)GO TO 40
      GO TO 35
   40 IF(PUL .NE. 672.)GO TO 41
      GO TO 35
   41 IF(PUL .EQ. 76B.)GO TO 35
   46 WRITE(6,47) NUMBD,CLASS,PLW,PLL,PUW,PUL
   47 FORMAT(1H0,10X,*BOARD NBR.*I4,* CLASS *R3,* HAS AN ERROR IN THE BO
     1ARD COORD. CARD*,2(2X,F5.0,2X,F5.0))
C--READ DEFECT COORDINATE CARD(S).
   35 DO 36 I=1,NDEF
      READ(5,20) DLW(I),DLL(I),DUW(I),DUL(I)
   36 CONTINUE
C--CHECK LWR COORD. LT UPR COORD. AND FOR DEFECTS OUTSIDE BRD COORD..
      DO 60 I=1,NDEF
      IF(DLW(I)) 56,52,52
   52 IF(DLL(I)) 56,54,54
   54 IF(DLW(I).GT.PUW) GO TO 56
      IF(DUW(I).GT.PUW) GO TO 56
      IF(DLL(I).GT.PUL) GO TO 56
      IF(DUL(I).GT.PUL) GO TO 56
      IF(DLW(I).GE.DUW(I)) GO TO 56
      IF(DLL(I).GE.DUL(I)) GO TO 56
      GO TO 60
   56 WRITE(6,57) I.NUMBD,DLW(I),DLL(I),DUW(I),DUL(I)
   57 FORMAT(1H0,10X,*DEFECT CARD*I3,* BUARD NBR.*I4,* IS IN ERROR*,
     12(2X,F5.0,2X,F5.0))
   60 CONTINUE
C--CHECK FOR OVERLAP OF DEFECTS
      IF(NDEF-1) 3,3,62
   62 J1=NDEF - 1
      DO 74 J=1,J1
      J2=J + 1
      DO 74 K=J2,NDEF
      IF(DUL(J)-DLL(K)) 74,64,64
   64 IF(DLL(J)-DUL(K)) 66,66,74
   66 IF(DUW(J)-DLW(K)) 74,6B,6B
   68 IF(DLW(J)-DUW(K)) 70,70,74
   70 WRITE(6,72) J,NUMBD,DLW(J),DLL(J),DUW(J),DUL(J)
   72 FORMAT(1H0,10X,*DEFECT CARD*I3,* BOARD NBR.*I4,* HAS OVERLAP*,
     12(2X,F5.0,2X,F5.0))
   74 CONTINUE
      GO TO 3
C
  100 CONTINUE
      CALL EXIT
      END
```

Program CTDFCT

```
      PROGRAM CTDFCT
     1(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE1)
C
C PROG TO WRITE BOARD DEFECT DATA CARDS ON TAPE(FILE 1) FOR INPUT TO PROG
C CONST. A DUMMY RECORD,CONTAINING 999,0,0,0,IS WRITTEN AS AN END-OF-BOARD
C INDICATOR. INPUT IS A SET OF DEFECT CARDS GROUPED BY BRD-WIDTH CLASS
C SUCH AS ALL 4-INCH, 8-INCH, OR 12-INCH BRDS.
C D.MARKSTROM  REPAIR STUDY  02/71  BJE
C
      DIMENSION LSTID(B)
C
      REWIND 1
      KDUMY=0
      KTREC=0
      KEND=0
      LAST=0
      NGRD=0
      NBRD=0
      NDEF=0
      IXS=0
      IYS=0
      IXE=0
      IYE=0
C--READ LIST ID CARD TO IDENTIFY OUTPUT (BA10)
      READ(5,100) LSTID
  100 FORMAT(BA10)
      WRITE(6,102) LSTID
  102 FORMAT(1H1,5X,BA10)
C
C READ BRD DEFECT CARDS AND OUTPUT TO TAPE1. LIST FOR MASTER RECORD.
C
   10 READ(5,104) NGRD,NBRD,NDEF
  104 FORMAT(5X,I2,24X,I5,32X,I4)
      IF(EOF(5)) 900,15
   15 LAST=NBRD
      WRITE(1,106) NGRD,NBRD,NDEF,KDUMY
  106 FORMAT(4I4)
      WRITE(6,10B) NGRD,NBRD,NDEF,KDUMY
  10B FORMAT(1H0,10X,4I5)
      KEND=NDEF + 1
      DO 20 I=1,KEND
      READ(5,110) IXS,IYS,IXE,IYE
  110 FORMAT(I3,1X,I3,5X,I3,1X,I3)
      WRITE(1,106) IXS,IYS,IXE,IYE
      WRITE(6,112) IXS,IYS,IXE,IYE
  112 FORMAT(1H ,10X,4I5)
   20 CONTINUE
      IXS=999
      IYS=0
      IXE=0
      IYE=0
      WRITE(1,106) IXS,IYS,IXE,IYE
      WRITE(6,112) IXS,IYS,IXE,IYE
      IXS=0
      KTREC=KTREC + KEND + 2
      GO TO 10
C
C END OF CARD INPUT. TERMINATE RUN.
C
  900 ENDFILE 1
      REWIND 1
      WRITE(6,114) LAST,KTREC
  114 FORMAT(1H0,5X,*END-OF-RUN. NBR OF LAST BOARD ON FILE = *I4,* TOTAL
     1 RECS WRITTEN ON FILE =*I5)
      CALL EXIT
      END
```

Erickson, Bernard J., and Donald C. Markstrom.
1972. Predicting softwood cutting yield by computer. USDA For. Serv. Res. Pap. RM-98, 15 p. Rocky Mt. For. and Range Exp. Stn., Fort Collins, Colo.

A computer program, written in FORTRAN predicts the maximum yield of cuttings for softwood cut-up and edge- and end-gluing operation. The program calculates cutting recovery (given cutting width and length constraints and defect locations on the board), and locates ripping saw kerfs.

Keywords: Programming (computers), softwood cut-up, end and edge gluing, dimension stock.

Erickson, Bernard J., and Donald C. Markstrom.
1972. Predicting softwood cutting yield by computer. USDA For. Serv. Res. Pap. RM-98, 15 p. Rocky Mt. For. and Range Exp. Stn., Fort Collins, Colo.

A computer program, written in FORTRAN predicts the maximum yield of cuttings for softwood cut-up and edge- and end-gluing operation. The program calculates cutting recovery (given cutting width and length constraints and defect locations on the board), and locates ripping saw kerfs.

Keywords: Programming (computers), softwood cut-up, end and edge gluing, dimension stock.

Erickson, Bernard J., and Donald C. Markstrom.
1972. Predicting softwood cutting yield by computer. USDA For. Serv. Res. Pap. RM-98, 15 p. Rocky Mt. For. and Range Exp. Stn., Fort Collins, Colo.

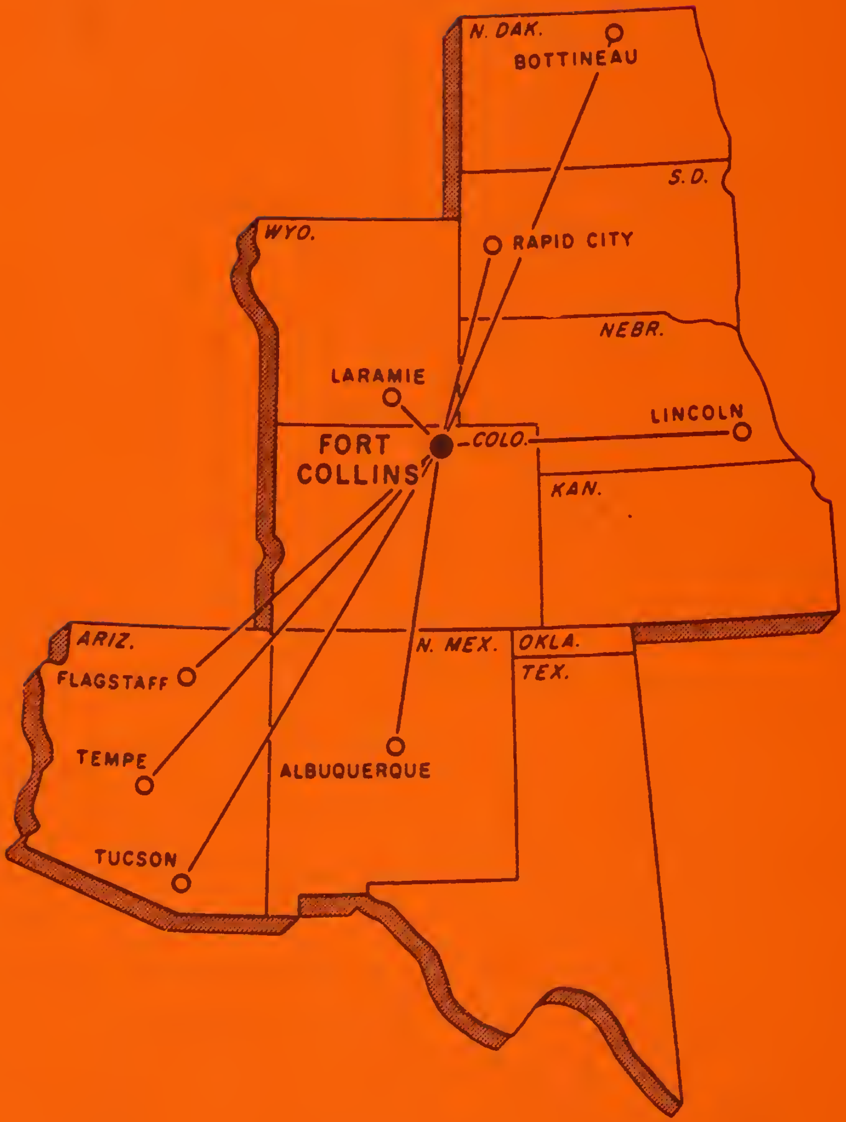
A computer program, written in FORTRAN predicts the maximum yield of cuttings for softwood cut-up and edge- and end-gluing operation. The program calculates cutting recovery (given cutting width and length constraints and defect locations on the board), and locates ripping saw kerfs.

Keywords: Programming (computers), softwood cut-up, end and edge gluing, dimension stock.

Erickson, Bernard J., and Donald C. Markstrom.
1972. Predicting softwood cutting yield by computer. USDA For. Serv. Res. Pap. RM-98, 15 p. Rocky Mt. For. and Range Exp. Stn., Fort Collins, Colo.

A computer program, written in FORTRAN predicts the maximum yield of cuttings for softwood cut-up and edge- and end-gluing operation. The program calculates cutting recovery (given cutting width and length constraints and defect locations on the board), and locates ripping saw kerfs.

Keywords: Programming (computers), softwood cut-up, end and edge gluing, dimension stock.


N. DAK.
BOTTINEAU
S.D.
WYO.
RAPID CITY
NEBR.
LARAMIE
LINCOLN
FORT COLLINS
COLO.
KAN.
ARIZ.
N. MEX.
OKLA.
TEX.
FLAGSTAFF
TEMPE
ALBUQUERQUE
TUCSON