AN ABSTRACT OF THE THESIS OF

James D. Anderson for the degree of Master of Science in Industrial Engineering presented on May 2, 1996. Title: Filling a Rough-Mill Cutting Order Using a Fuzzy Logic Controller.

Abstract approved: _Redacted for Privacy

Charles C. Brunner

Abstract approved: _Redacted for Privacy

Sabah Randhawa

Because of their high cost, reducing the number of boards used to fill an order of dimension parts is an important goal in a lumber rough mill. FLCO (Fuzzy Logic Cutting Order) is a computer program that achieves this goal. Using the concepts of fuzzy logic control, FLCO provides a heuristic approach to this problem. FLCO incorporates a version of the CORY lumber cut-up software, and provides a model of a rough-mill system that is able to reduce the number of boards needed to fill a cutting order. FLCO allows different control methods to be used, including fuzzy logic control, dropping sizes from the cutting order as their demands are met, and no control. Because of the modularity of its design the FLCO code can easily incorporate other control methods.

CORY lumber cut-up software provides sawing solutions for boards based on the values assigned to the sizes in a cutting order. After each board is sawn, the fuzzy logic controller adjusts the value of each size to achieve the objective of filling the demand for each size in a cutting order at about the same time.

These new values are subsequently used by CORY when finding a sawing solution for the next board to be processed. Upon completion, FLCO reports the number of boards required to fill the cutting order, the average percent area yield, and the number of each piece recovered.

Three cutting orders, three lumber grades and two types of control provide a means of determining the effects of fuzzy logic control on reducing the number of boards used to fill a cutting order. Across all lumber grades and cutting orders, fuzzy logic control greatly reduces the number of boards as compared with dropping sizes. Fuzzy logic control slightly reduces the number of boards as compared with a more complicated method of control across all lumber grades and most cutting orders, but the difference is not statistically significant. In addition, the effect of fuzzy logic control on individual sizes is also examined.

Filling a Rough-Mill Cutting Order Using a Fuzzy Logic Controller

by

James D. Anderson

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

Presented May 2, 1996
Commencement June, 1996

Master of Science thesis of James D. Anderson presented May 2, 1996

APPROVED:

Redacted for Privacy

Co-Major Professor, representing Industrial Engineering

Redacted for Privacy

Co-Major Professor, representing Industrial Engineering

Redacted for Privacy

Head of Department of Industrial and Manufacturing Engineering

Redacted for Privacy

Dean of Graduate School

I understand that my thesis will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my thesis to any reader upon request.

Redacted for Privacy

James D. Anderson, Author

ACKNOWLEDGEMENTS

I would like to express my deep gratitude to Dr. Charles C. Brunner, for serving as co-major professor, and acting as an informal advisor and instructor during the time I have worked with the computer-aided manufacturing group in the Forest Products Department. Without his generous help, advice and encouragement this thesis would not have been possible.

Many thanks to the other members of the computer-aided manufacturing group, including Dr. James W. Funck, Dr. David A. Butler, Dr. Johannes B. Forrer and especially Mr. Alberto G. Maristany for giving me their support and sharing their experience and knowledge.

Thanks are also due to Dr. Terrence G. Beaumariage and Dr. David G. Ullman for their effort and advice while serving on my graduate committee. I owe special thanks to Dr. Kenneth H. Funk II for his role both as an instructor and minor professor and Dr. Sabah Randhawa for his time and expertise acting as a co-major professor, graduate advisor and instructor during the course of my studies.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

Page

TABLE OF CONTENTS (Continued)

LIST OF FIGURES

LIST OF TABLES

FILLING A ROUGH-MILL CUTTING ORDER USING A FUZZY LOGIC CONTROLLER

1. INTRODUCTION

Cutting dimension parts from lumber is an important process in several industries, including molding and millwork production, pallet production and furniture manufacturing, to name a few. Although the details of the procedure vary from mill to mill, the raw material is always rough-sawn lumber that is sawn further by an assortment of ripcuts (which approximately parallel the grain of the wood) and crosscuts (which are perpendicular to the ripcuts). The rips and crosscuts partition each board into a set of rectangular portions and are considered either as waste or as pieces with some value to an end-user. A portion of each board is considered waste when its dimensions are not wanted, or when it contains natural features[1] that are unwanted for a particular application. Inversely, if a portion of a board has dimensions that are desirable and if its features are allowable by a user, that portion has some value associated with it and can be used by a customer in an application. The goal of rough-mill production is to obtain these valued pieces from the input raw material.

[1] Often these features are referred to as "defects." However, some features that are considered defects in one application may be allowable or even desired in another application. In this paper, the terms "feature" and "characteristic" are preferred over "defect." Portions of wood that are free from disallowable features are referred to as "clear areas."

1.1 Rough-Mill Sawing Processes

Several procedural parameters characterize rough-mill sawing processes and are shown in Table 1.

Table 1 Rough-mill procedural parameters

Parameter	Options
Board edging	Amount of edging or unedged
Type of first sawing operation	Crosscut or rip
Number of sawing stages	Two or more
Piece dimensions	Fixed or random at each stage
Kerf width	Crosscut and rip at each stage
Number of operations	One or more at each stage
Blade setup	Blade movement limitations and elemental or non-elemental pieces

1.1.1 Board edging, type of first sawing operation and number of stages

Boards sawn into dimension parts may be edged or unedged. Edging removes some board material along its length, and is often done to remove bark or missing wood on its edge (Schott, 1995). The first type of sawing operation applied to each board in a mill is either a crosscut or a rip. Usually a board will undergo several crosscuts or several rips before a change of operation takes place. The set of operations between these changes is termed a stage (Anderson et al., 1992), and at least two sawing stages are applied to a board in any mill setup: crosscuts followed by rips or rips followed by crosscuts. Figure 1 is a state diagram for a three-stage rip-first process. Boards are processed sequentially, and at each stage some material from the board may go to waste. Other portions are either processed further, or considered finished parts.

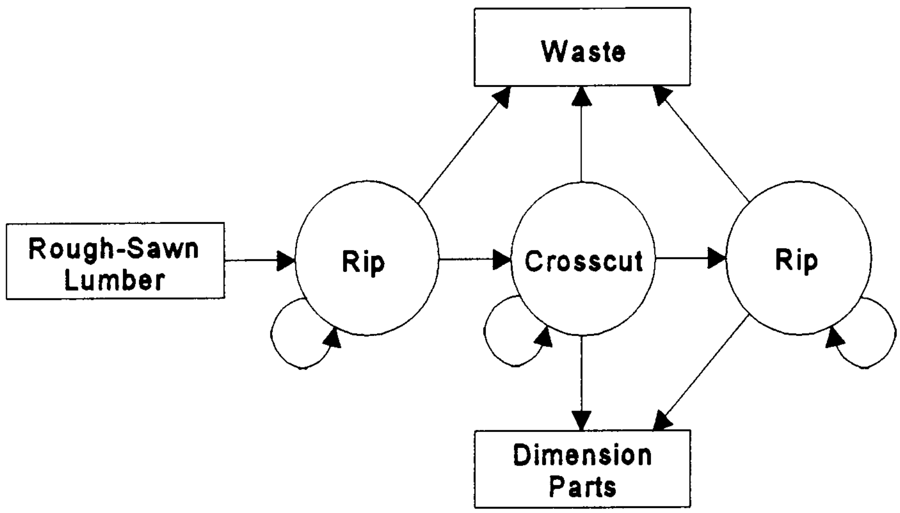

Figure 1 State diagram for a three-stage rip-first sawing procedure

1.1.2 Piece dimensions

At each stage, the portions sawn may be of either fixed or random dimensions, depending on the particular process. In a three-stage rip-first scheme as illustrated in Figure 1, a common approach is to allow both fixed and random dimension lengths to be cut at the second stage. The fixed length pieces are usually of dimensions specified by some customer, while the random length pieces are finger-jointed together for some application where the appearance of joints in the material does not matter.

1.1.3 Kerf width

Another parameter that is relevant in the rough-mill environment is the width of the saw kerfs, which is the board material removed by the saw blades. This parameter is closely related to the physical limitations of the blades, and is not easily changed by the mill operator. However, proper maintenance can decrease variability in kerf and improve the process (DeVor et al., 1992), which can lead to increased recovery (Rahardjo, 1992).

1.1.4 Number of sawing operations and blade setup

The number of sawing operations that can be done at each stage may depend on the type of equipment in place. For example, a first-stage rip saw will often have four to six circular blades mounted in parallel on a shaft, and in the first stage all of the rip operations on a board are done simultaneously. By comparison, a second-stage crosscut saw often has only a single blade perpendicularly mounted to the length of the board. The blade is lowered to make cuts as the wood is moved past it. Only a few physical constraints limit the number of possible crosscut operations. This is closely related to the blade setup, which describes the limitations in movement (if any) of the blades relative to the board, and whether or not the resulting pieces are required to be of a dimension specified by a customer. If all of the useful pieces produced at a given stage possess at least one dimension specified by a customer, they are called elemental pieces (Anderson et al., 1992). Non-elemental pieces are never produced by the last two stages of any rough-mill process; the last two stages provide the final opportunity for pieces to be cut to a dimension wanted by an end user.

1.2 Rough-Mill Input

The parameters previously described relate to the actual processing of rough-sawn lumber. Board quality and cutting order are two more important limitations which can be considered separately because they relate more to the input to the rough-mill process than the processing itself.

1.2.1 Boards and board quality

The quality of the boards used in a mill will affect the ease with which obtaining the desired dimension parts is achieved. For example, obtaining large dimension pieces from low quality boards is more difficult than from high quality boards. Grades describe board quality. Grades, industry adopted standards, categorize boards mostly by minimum size requirements and by maximum frequency and size of certain features (WWPA, 1988), or by clear area size requirements (NHLA, 1986). A mill manager can somewhat determine the quality of boards being cut by choosing to use boards of a certain grade[2]. However, quality variation can occur within grades because a lumber grade is a discrete, not continuous, measurement of quality. Boards, with their natural variation, range from those that barely meet the requirements for one grade to those that barely fail the requirements for the next higher grade. The mill manager cannot control this within-grade variation (Carino and LeNoir, 1988).

1.2.2 Cutting order

In its simplest form, a cutting order is a list of piece sizes, the number of each piece required by a customer, and an associated value. Frequently a specification of allowable and disallowable features will also be associated with

---

[2] Limited availability of certain board grades or other constraints may influence the quality of the boards being cut as well.

each piece. A mill manager must produce the requested items in the cutting order. Like board quality, a cutting order is an external constraint imposed on the rough-mill process. Certain cutting bills are more constraining than others; generally those that contain many large sizes or have extremely high demands are more difficult to meet than those that do not.

1.3 Problem Statement

Corresponding to recent increases in the cost of lumber is an increase in the importance of extracting the most value from the lumber used by a mill. Carino and Foronda (1990) state that "on the average, lumber input cost constitutes about 40 to 60 percent of the total cost in producing hardwood furniture." Similar percentages are cited for cabinet production (Carino and LeNoir, 1988). If these industries reflect trends in other areas of lumber processing, then raw material is a dominating cost in the manufacture of dimension parts. Therefore, an important way to reduce the overall cost of producing dimensional material is to reduce the amount of lumber used.

Using less lumber in mill production provides several other benefits. One of these is lower material handling cost. Another is that cutting orders are filled efficiently, allowing a mill to meet more customer orders. Further, less material goes either to waste or to low valued products such as chips. Finally, fewer cut trees are needed, saving an increasingly scarce natural resource.

The problem for mill managers can be stated as follows: how can the cutting order be filled while using as few boards (or as few board feet) as possible? This is an instance of a two-dimensional cutting stock problem, subject to several constraints that should be noted. First, the stock material (ie. the boards) contains features that are not allowed in the final product, and whose location is not easily predicted. Second, the stock material's initial sizes are not known. Third, allowable cuts will be either parallel or perpendicular to every other cut, and each will be guillotine cuts (Gilmore and Gomory, 1966). Fourth, because wood is an anisotropic material, piece orientation is important.

2. LITERATURE REVIEW

Within the problem area considered here, two categories of optimization can be identified. Carino and Foronda (1990) call these local and global optimization. In this paper, local optimization involves cutting up boards in the best possible manner, while global optimization involves finding the least-cost solution to meeting the customer's cutting order (Dmitrovic et al., 1992). In the literature, most approaches concentrate on one or the other of these. Classifying some references either under the heading "Local Optimization" or under "Global Optimization" is somewhat artificial, but is helpful for identifying the features most relevant to this paper. Both local and global optimization can be considered under the general category of cutting and packing problems, which have been the subject of considerable research.

2.1 Problem Typology

Sweeney and Paternoster (1991) found more than 400 papers, proceedings and theses on cutting and packing problems. Part of the reason for the interest in them lies in their wide range of applicability, including areas as diverse as bin packing, vehicle loading, partitioning problems, line balancing, multiprocessor scheduling, and even capital budgeting. This wide variety prompted Dyckhoff (1990) to propose a typology of cutting and packing problems based on four characteristics common to all:

1. Dimensionality of stock objects

    (1) One-dimensional.

    (2) Two-dimensional.

    (3) Three-dimensional.

    (*N*) *N*-dimensional with $N > 3$.

2. Kind of assignment

    (B) All stock objects and a selection of items.

    (V) A selection of stock objects and all items.

3. Assortment of large stock objects

    (O) One stock object.

    (I) Identical figure (ie. shape).

    (D) Different figures.

4. Assortment of small items

    (F) Few items (of different figures).

    (M) Many items of many different figures.

    (R) Many items of relatively few different (non-congruent) figures.

    (C) Congruent figures.

Under this typology, "items" refers to the pieces or products placed in or cut from the stock objects. The problem considered in this paper can be classified as 2/B/D/M: a two-dimensional problem where all objects (boards) are assigned a selection of items (pieces), the objects are of different figures into which many items are to be placed.

2.2 Local Optimization

While some references in this section discuss issues of global optimization, they are reported here because their emphasis is on local optimization. They are more concerned with placing pieces in a stock object than with the overall goal of obtaining the required number of pieces.

2.2.1 Placement problems

Although the cutting stock problem was described much earlier, (Kantorovich, 1960; Eisemann, 1957) Gilmore and Gomory's work on cutting and packing problems from the 1960's (Gilmore and Gomory, 1961, 1963, 1965) provided practical solution techniques and a foundation for much of the research that has followed. Gilmore and Gomory (1965) briefly explain how to extend their techniques to account for character variations in the stock material.

Hahn (1968) further develops Gilmore and Gomory's ideas, using dynamic programming to find optimal three-stage cutting solutions for two-dimensional areas with disallowable features. Hahn simplifies the problem by allowing only guillotine cuts, and assumes that all items recovered at the third stage are the same size. The algorithm Hahn describes is not constrained by the number of pieces required, although she suggests that piece values should be set considering inventory and demand requirements. The computation time for the algorithm increases linearly with the number of defects, and more than quadratically with the number of sizes. This is not as serious of a problem today as then, but the large computation times have provided a motivation for the development of faster solution techniques (Pegels, 1967; Herz, 1972; Adamowicz and Albano, 1976).

Pegels (1967) compares two heuristic models that solve a 2/B/I/R problem, where the stock material is without quality variations. Herz (1972) gives a recursive technique for solving 2/B/ / problems that is both fast and optimal. The technique assumes isotropic, stock material of uniform character. Christofides and Whitlock (1977) present a non-recursive method for the same class of problems as Herz. They use a depth-first branch and bound strategy, and limit the number of times a piece can appear in a single stock object.

A method to optimize guillotine cutting of unedged boards with disallowable features is described by Scheithauer and Terno (1988). They improve on the computational efficiency of Hahn by eliminating some unnecessary calculations, but like Hahn, they do not consider the demand for each piece.

Brunner et al. (1990) revise the Gilmore and Gomory algorithms specifically for determining optimal sawing patterns for board clear areas. One simple change they make is to consider the thickness of kerf lines. More importantly however, they overcome some memory requirements of the algorithms by eliminating redundant solutions from consideration, and by using efficient data structures. Further, they take advantage of the fact that clear areas in boards that are both very long and very wide occur infrequently, and thus do not find solutions for them. By using this approach, they significantly reduce the computation time required.

Carnieri et al. (1993) report a heuristic procedure for cutting lumber. They report that it provides optimal or near-optimal solutions. The procedure has limited applicability because it can only be used for lumber with a single disallowable feature. They claim it can be extended to include more features, but offer no hints on how to do so.

Rönnqvist (1995) describes a method for single-stage crosscutting of wood strips into pieces with desired lengths and qualities. His approach addresses a situation which often arises in mills: different end-products often have different quality requirements. To satisfy the real-time constraints of the problem, he makes some simplifying approximations, but still arrives at solutions which are near-optimal.

2.2.2 Lumber cut-up programs

Several computer-based implementations of placement models exist. Those most relevant to this paper are those that model lumber cut-up operations. The earliest of these is Thomas' rough-end yield program (1962). Since then, many have been developed, including YIELD (Wodzinski and Hahm, 1966), RIPYLD (Stern and McDonald, 1978), MULRIP (Hallock and Giese, 1980), OPTYLD (Giese and McDonald, 1982), CROMAX (Giese and Danielson, 1983), CORY (Brunner, 1984; Brunner et al., 1989), GR-1ST (Hoff et al., 1991), AGARIS (Thomas et al., 1994) and ROMI-RIP (Thomas, 1995b). These programs have several characteristics in common. They all solve 2/B/D/M problems with the additional constraints of guillotine cuts and disallowable features in the stock material. Any differences between these programs are primarily in their modeling of the rough-mill procedural parameters listed in Table 1 on page 2.

2.3 Global Optimization

While some references in this section discuss issues of local optimization, they are reported here because their emphasis is on global optimization. They are more concerned with the overall goal of obtaining the required number of items than with placing items in an object.

2.3.1 Cutting stock problems

Like Herz, (1972) Adamowicz and Albano (1976) also give a technique for 2/B/ / problems. Their heuristic approach is similar to Herz's in that it assumes the stock material is of uniform character. It is different in that piece orientation is important, the number of pieces to be cut is limited, and the cuts are non-guillotine. Albano and Orsini (1980) extend the approach to consider guillotine cuts, but still assume feature-free stock.

Cheng and Pila (1977) use dynamic and integer programming to maximize the use of stock material, with a focus on lumber. They model a two-stage crosscut-first system that allows random width pieces, and considers the presence of features in the stock. As they state it, their "objective is to cut a large number of boards to satisfy a cutting [order] with an overall gain that is to be increased to a maximum." Their approach to the problem is to make sure that "the ratio between the number of pieces . . . to the original required quantity is approximately the same for each and all items." In other words, they attempt to meet the demand for all pieces at about the same time. They dynamically assign values to a piece based on the percent demand met for that piece at a given instant of production. Any piece with a percentage exceeding the average ratio for all pieces is not cut.

A heuristic procedure for one-dimensional cutting of lumber stock is given by Azarm et al. (1991). Rather than evaluating their method with actual board data, they use Monte-Carlo simulation to generate the boards and the features they contain. They observe that longer pieces are more difficult to recover than shorter pieces, and to account for this they dynamically assign priorities, or values, to each piece based on how fast the piece is being recovered.

Dmitrovic et al. (1992) develop a model for lumber cut up which considers both local and global optimization. They recognize that in an industrial setting, little is known about lumber quality before cutting, and argue that therefore homogeneous quality must be assumed. Therefore one cannot give a good reason to produce one piece in preference to another, and so they maintain "a relative constant proportion between all the pieces: at x% of production, it is required that x% of the quantities requested be complied with." Like Cheng and Pila, they attempt to meet the demand for all pieces at about the same time.

Carnieri et al. (1994) report some algorithms for cutting dimension parts from lumber (anisotropic stock) or composite boards (isotropic stock). The procedure can be used for either crosscut or rip-first systems, and even determines the best first operation. Like Carino and Foronda's SELECT program (1990), their methods require detailed inventory information.

2.3.2 Cutting stock programs

Cheng et al. (1977) use the methods described by Cheng and Pila (1977) in a benchtop model of an automated lumber sawing system. The optimization routines dynamically allocate values to pieces in the cutting order to maximize the use of stock material.

OPTIGRAMI (Martens and Nevel, 1985) is a lumber allocation model that determines least-cost grade mixes of lumber. It uses linear programming to fill a cutting order with a user-specified selection of graded lumber. OPTIGRAMI can consider any volume limitations for each grade being used.

SELECT is another lumber allocation model, developed by Carino and Foronda (Carino and Foronda, 1990; Foronda and Carino, 1991), which accounts for quality variations in the stock material. Using both linear and nonlinear programming techniques, it requires that the length, width, grade and number of each board in inventory are known beforehand, a condition that is impractical. The objective of SELECT is to find the least-cost allocation of lumber required to fill the cutting order.

Recognizing that a mill production environment is often changing, Voigt (1987) has developed software to solve the cutting order problem that allows for computer-aided, but human controlled cutting pattern generation. Voigt gives several realistic but problematic conditions most of which are not considered by programs like OPTIGRAMI and SELECT:

- fast changes in production programs
- short series of production
- discontinuous delivery of boards
- differences in quality and type of boards.

Rather than trying to account for these conditions with software, Voigt's program can either assist a human operator in generating sawing patterns, or allow for human-generated solutions.

3. APPROACHES, OBJECTIVES AND SCOPE

As indicated by the wealth of literature on the subject, many potential approaches to the problem of filling a cutting order in secondary lumber manufacturing exist. These can be roughly divided into two categories: heuristic and non-heuristic.

3.1 Heuristic and Non-Heuristic Methods

Generally, heuristic methods are those that approach a "good" solution to a problem without guaranteeing that the solution found is the "best." The true maximum (or minimum) may be too difficult or costly to obtain. Heuristics are methods that approach a maximum (or minimum) solution more easily or cheaply.

The simplest example of a non-heuristic method to solve the problem stated in Chapter 1 is no control, which is the equivalent of cutting pieces for inventory. This strategy has the advantage of offering high yields, but has the disadvantage that many pieces may be cut for which there is no demand. Other non-heuristic methods that do apply some sort of control are usually based on a mathematical model of the problem. These methods do not have the disadvantage of no control, but they do have the disadvantage of being difficult or costly to carry out.

An example of a heuristic method is that of dropping sizes from the cutting order as the demand for that size is filled. This has the advantage of simplicity, but has the disadvantage that no priority is given to pieces that may be difficult to obtain from the boards that are being cut. Smaller pieces that are easy to recover, or pieces with low demand will meet their demands very rapidly and be dropped from the cutting order. Larger pieces that are more difficult to obtain, or pieces with large demands will remain in the cutting order longer. This results in there being progressively fewer pieces in the bill, while pieces that remain become harder to recover. Therefore yields will usually drop as cutting progresses. Other heuristic methods include dynamically changing piece values in response to factors such as piece sizes, demands and recovery rates.

3.2 Justification for Using Fuzzy Logic Control

Fuzzy Logic Control (FLC) is an example of a heuristic method that can be used to change piece values dynamically as cutting progresses, thus affecting the number of each piece recovered by a sawing system. While other methods of control may be used to provide reasonable solutions for the problem area under consideration, FLC is a suitable alternative. Cox (1992) gives four conditions under which using fuzzy logic for control is appropriate. First, FLC is appropriate when one or more of the control variables are continuous. This condition is satisfied for all of the control variables, as will be shown later (see discussion of control variables on page 25).

Second, FLC is appropriate when a mathematical model of the process does not exist, or is too complex to be used in the controlled system under consideration. Presently there is no mathematical model of the process, but one is being developed (Hamilton, 1996).

Third, FLC is appropriate when high ambient noise levels are present. In the rough mill environment, one factor contributing to the "noise" is variability of board quality; a board of any quality may be cut at any time (Carino and LeNoir, 1988). In the most extreme case, this could mean that a board of the highest grade could be followed by a board of the lowest grade, or vice-versa. Most of the time though, a board's quality is probably similar to the quality of the boards around it. However, this noise factor is still present due to within-grade variation of board quality.

Fourth, FLC is appropriate when expert knowledge is available to specify the rules underlying system behavior and the fuzzy sets that represent the behavior of the variables. Here, expert knowledge is available in the person of the author, with seven years of experience with CORY (Brunner, 1984; Brunner et al., 1989) sawing model software.

3.3 Project Objectives and Scope

The purpose of this project is to solve the instance of the cutting stock problem that exists in the manufacture of dimension parts from boards. The primary objectives of this research were to:

1) develop a Fuzzy Logic Controller to solve this problem, using the CORY software to generate sawing solutions;
2) compare the performance of the FLC with other methods of control, noting especially their effect on lumber usage requirements.

A secondary, but important objective was to develop software that provides a user interface and integrates the CORY and FLC modules.

Some assumptions limit the scope of the project:

1) the cost of stock material dominates all other costs to the manufacturer;
2) boards are processed sequentially, and board quality is not known before sawing;
3) piece sizes will not be added to the cutting order during board processing;
4) all pieces cut will be of the same quality.

The first assumption implies that other approaches to reducing mill costs are not considered here. The second assumption relates to the timing and mix of board grades and the third assumption to the timing and mix of cutting orders. Both of these timing and mixture problems have been explored elsewhere (Hafley and Hanson 1973; Chambers and Dyson, 1976; Martens and Nevel, 1985; Carino, 1986; Carino and LeNoir, 1988; Carino and Foronda, 1990; Farley, 1990; Foronda and Carino, 1991), and are beyond the scope of this work. The fourth assumption relates to the fact that mills will often specify pieces in terms of their quality as well as their size, value and demand. Quality variation in the pieces cut is explored by Rönnqvist (1995), and is also beyond the scope of this work.

4. FUZZY LOGIC CUTTING ORDER MODEL DEVELOPMENT

Figure 2 shows the parts of a general control system. The subject of control might be a physical device such as a voltage regulator, or it may be more abstract, such as a set of prices. In either case, the state of the process is used as input for the decision making logic, and the output of the decision making logic is used to change the behavior of the controlled system.

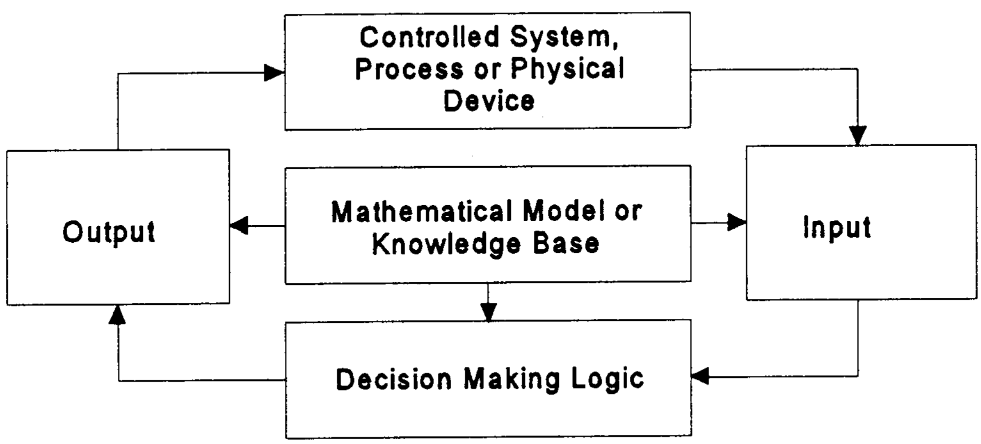

Figure 2 General control system

The center and bottom boxes in Figure 2 illustrate that the decision making logic relies either on a mathematical model, or on another representation of information. These two boxes stand for the expert knowledge contained in the system. As the name implies, decision making logic tells the system what decisions to make--what aspects or what variables of the controlled system need to be changed. Alongside this, the mathematical model or knowledge base tells the system how to make those decisions--how much to change the variables of the controlled system. The knowledge base also provides a means by which the process and the decision engine can exchange information that is meaningful to each another, for example by converting raw physical data into numerical measurements.

A fuzzy control system can be represented very similarly as shown in Figure 3 (Lee, 1990a, 1990b; Cox, 1992; Mendel, 1995), which differs from a generalized control system primarily in the addition of fuzzification and defuzzification interfaces. The fuzzification interface converts state information about the controlled process into their fuzzy representations. Inversely, the defuzzification interface changes the fuzzy results of the decision making logic into crisp control values. Notice that all decisions are made using fuzzy information. This means that all of the decision making logic uses fuzzy sets and fuzzy rules of inference, which are explained in Appendix 2.

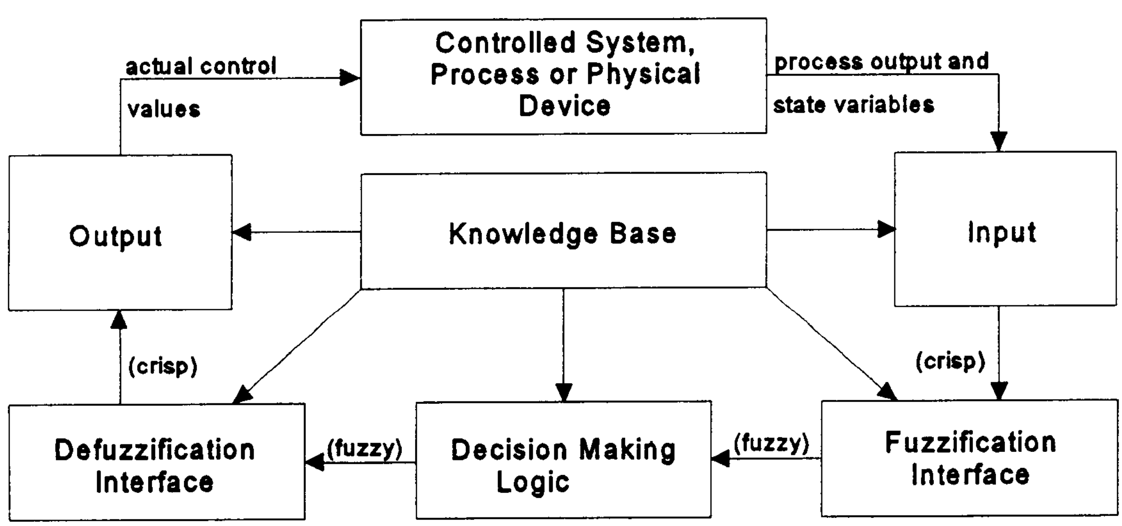

Figure 3 Fuzzy control system

Most of the boxes in Figure 3 correspond to some program module in a fuzzy logic controlled system. The knowledge base is an exception, and includes knowledge about input and output to the FLC and knowledge about the appropriate fuzzy sets and fuzzy rules for a particular application. The knowledge base represents all of the expert knowledge embodied in the system and is not so easily modularized.

As stated on page 17, the purpose of this project is to develop a Fuzzy Logic Controller to solve the problem of filling a cutting order in a modeled rough-mill environment. FLCO (Fuzzy Logic Cutting Order) is a software package designed to meet this goal. Figure 4 shows the major portions of the system. Global optimization, which here corresponds to obtaining the required

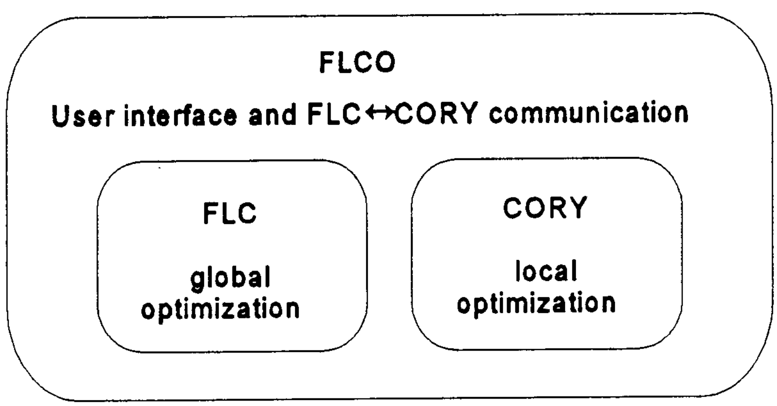

Figure 4 Major portions of FLCO software system

numbers of dimension parts, is done by a fuzzy logic controller, developed specifically for this purpose. Local optimization, which here corresponds to finding sawing solutions for individual boards, is done by a version of CORY (Brunner, 1984; Brunner et al., 1989), a sawing process model. Since local optimization is not the focus of this project, a library of CORY routines provides a programmer interface via a few function calls. This library can easily be replaced by any other CORY sawing model, or by a different sawing algorithm altogether. The local and global optimization modules are contained within the larger FLCO program that enables communication between them and provides a graphical user interface.

To describe the FLCO system, this chapter is organized to reflect the schematic representation of a fuzzy logic controlled system in Figure 3 on page 20. First is a description of the controlled process: the CORY sawing model.

4.1 CORY Sawing Model

The CORY family of programs model a variety of sawing processes, differentiated by the rough-mill procedural parameters[3] they embody. A heuristic decision engine drives each of them, and determines the best placement of cuttings for individual boards.

4.1.1 Description of the algorithm

As input, CORY requires two items: board data and a cutting bill. A set of Cartesian coordinates and feature codes represent a board. The coordinates specify the board's boundaries and the location of any noteworthy features within the board. Feature codes detail the type of each feature, and the face of the board on which they appear. More details about board data requirements can be found in Brunner (1984).

CORY finds a sawing solution for an individual board by choosing the kerf line locations that result in the highest value of parts recovered. Rather than examining all possible positions of kerf lines, CORY considers several positions that are likely to be among the most valuable. This provides a tremendous decrease in execution time with little reduction in recovery (Brunner and Anderson, 1991).

---

[3] Table 1, page 2.

To evaluate these kerf line locations, CORY considers each board as a collection of clear areas, where a clear area is a rectangular region within the board that is free of disallowable features. Once these clear areas are found, CORY assigns each the value of the most valuable cutting that can be placed within it. For example, consider a clear area of dimensions 41" x 2.25", and a cutting bill containing sizes 40" x 2" and 39" x 1.75", valued at 80 and 85 respectively. The clear area is assigned a value of the smaller size, because CORY makes decisions based on cutting value, not area.

Assigning a value to a clear area based on a single cutting is not the usual approach used by the CORY family of sawing models. Usually, the optimal pattern of cuttings that will fit in a clear area defines its value (Brunner et al., 1990). This clear area optimization represents a third level of optimization in addition to the local and global optimization already discussed. To avoid too many complicating factors in the development of the FLC, the CORY sawing model used for this project is modified to assign the value of a single cutting to each clear area, not the value of an optimal cutting pattern.

After all clear areas have been given a value, CORY determines the worth of each kerf line location by the total value of the clear areas that can be recovered once the board is sawn at that location. The board is then "sawn" along the most valuable kerf line, and repeats the entire process on the resulting portions until the entire board is reduced either to waste or to dimension parts.

4.1.2 Description of procedural parameters modeled

For this project, the version of CORY used closely models a sawing system that is commonly used in the Pacific Northwest for producing molding and millwork--a three-stage rip-first system. Either edged or unedged boards may be used, and only fixed dimension pieces are recovered at each stage. In a mill setting, it is more common to cut both fixed and random length pieces during the second stage. However, since random length pieces are glued up into finger-joint material, they do not have a specific demand count associated with them as fixed length pieces do. They are generally cut to inventory, and the problem addressed in this paper does not apply to them. Further, fixed length pieces are preferred to similar sized random length pieces because they are usually more valuable. Random length pieces are often recovered from the clear areas that are "left over" in a board after the fixed length pieces have been recovered. For these reasons, and without loss of generality, the CORY sawing model used here does not consider random length pieces.

Constraints on the primary ripsaw operations model the mechanical limitations of an actual ripsaw produced by an Oregon company. The ripsaw has six movable blades, each of which has a maximum range of movement of 8 and 13/16 inches, and can be positioned accurately to the nearest 1/1000 of an inch. Primary rip blades cannot be positioned closer than 7/8 inches to any other primary blade. All strips resulting from the primary ripsaw operation are either waste, or widths that are present in the cutting order. Because of the limited number of blades, up to six rip operations can be made at the first stage. Any number of second stage crosscuts can be made on each strip, and at the third stage saw, at most one re-rip can be made on each section. All kerf widths are 1/8 inches. Table 2 summarizes the procedural parameters modeled by CORY in the FLCO application.

4.2 Fuzzy Logic Controller

For more detail about the operation of the FLC, please refer to Appendix 2. Referring again to Figure 3, one can see that the controlled system, CORY, needs to be given some information to control its behavior, and needs to provide some information to describe its state. CORY's output becomes the input for the FLC, and the FLC's output becomes the input for CORY. In this way, after each board is processed, the FLC assesses the current state of CORY's performance and how well CORY is approaching the overall goal. The FLC uses this information to make changes to some of CORY's input parameters, thus changing CORY's performance on subsequent boards. Any parameters used as input to or output from an FLC system are known as its control variables.

Table 2 CORY model procedural parameters

Parameter	Setting
Board edging	Unspecified
Type of first sawing operation	Rip
Number of sawing stages	Three
Piece dimensions	Fixed dimensions at all stages
Kerf width	All kerfs 1/8 inch
Number of operations at each stage	First stage - up to six Second stage - no limit Third stage - zero or one
Blade setup	Up to 8 and 13/16 inches movement for each primary rip blade. No limitations on any other blade. Elemental pieces only at each stage.

4.2.1 Control variables: FLC output and input

Since CORY makes sawing decisions based on the values of the sizes in the cutting order, changes to those values will cause changes in CORY's sawing decisions. Therefore, changes in size values are used as output from the FLC.

Since the goal of the FLC is to meet the demand for each size in the cutting order with as few boards as possible, the input to the FLC is the number of cuttings of each size recovered from every board CORY processes. CORY can provide recovery information aggregated over all boards it processes.

However, to minimize its interface with the FLC, only a list of sizes recovered is returned for each board processed. The supporting FLCO software has the responsibility to keep track of the total number of boards processed and the total number of pieces recovered of all sizes. A list of cutting sizes is the most rudimentary form of recovery information, and from it the FLC gathers what it needs to make control decisions. To do this, it needs an objective.

4.2.2 FLC objective

In a control application, the purpose of the controller is to keep the controlled process operating within some reasonable boundaries. Usually a specific control value exists for every parameter that is being managed. This is true for most FLC's as well. In the case of the FLCO application, the FLC attempts to meet the demand for all of the sizes in the cutting order at about the same time. As Dmitrovic et al. (1992) state, "without *a priori* knowledge of the quality of the raw material to be processed, the ideal would be to apportion production of the pieces equally over the entire production in view."

For the FLCO application, the FLC needs a specific numeric control value (or objective). The control value to which the FLC aspires is the average expected number of boards required to fill the cutting order. If the expected number of boards required to meet the demand for an individual size is larger than the average, that size's recovery is slower than most other pieces in the cutting order. Similarly, if the expected number of boards required to meet the demand for a size is smaller than average, it's recovery is faster than most. By increasing the value of the former sizes, and decreasing the value of the latter, the FLC can influence CORY's recovery of them and distribute their production more evenly.

After processing each board, FLCO calculates the expected number of boards needed to meet demand for each size in the cutting order and averages them to find the value of the control objective. This control value is averaged with the control value calculated for the previous board to keep the FLC from taking too drastic control actions. Consider the following variables:

i	index of sizes in cutting order, i = 1, 2, . . . , n;
j	index of boards cut, j = 1, 2, . . . , m;
n	number of sizes in cutting order
m	number of boards cut
s_i	i^{th} size in cutting order;
B_j	j^{th} board cut;
d_i	demand of s_i;
q_{ij}	quantity of s_i recovered after cutting j boards;
r_{ij}	q_{ij} / j; rate of production (in pieces per board) of s_i after cutting j boards;
e_{ij}	d_i / r_{ij}; expected number of boards required to meet demand for s_i after cutting j boards;
A_j	average expected number of boards required to meet demand for all sizes after cutting j boards;
G_j	weighted average expected number of boards required to meet demand for all sizes after cutting j boards.

A_j is calculated by the following:

$$A_j = \frac{\sum_{i=1}^{n} e_{ij}}{n}. \tag{1}$$

Upon receiving a sawing solution from the CORY software, FLCO calculates the expected number of boards required to meet demand for every size in the cutting order. A_j is the arithmetic average of those expectations. G_j is recursively defined by:

$$G_j = \frac{A_j + G_{j-1}}{2}, \quad where\ G_1 = A_1. \tag{2}$$

FLCO calculates G_j after each board is sawn, and is the objective value to which the FLC aspires. It is the current value of A_j averaged with the objective value found after sawing the previous board. By averaging these two, FLCO prevents the objective value from changing too drastically in response to drastic changes in quality between subsequent boards.

Usually, the FLC increases the value of every size i for which e_{ij} is greater than G_j, and decreases the value of every size i for which e_{ij} is less than G_j. This is described in more detail under the section on decision making logic. The net effect of this method is that demands for all sizes in the cutting order are filled at approximately the same time. Dmitrovic et al. (1992), Cheng and Pila (1977) and Azarm et al. (1991) also use this heuristic approach to reduce the number of boards needed to fill a cutting order.

Behind this method is the intuitive idea that a cutting order with many sizes allows for more efficient use of individual boards than a cutting order with few sizes. This idea is sensible because not all boards possess large enough clear areas to accommodate all sizes in a cutting order. By keeping the number of cutting sizes as large as possible, the likelihood that a board will contain a clear area large enough to accomodate one of them is increased, thus improving board usage. If individual boards are used efficiently, then a cutting order can potentially be filled with fewer boards than if individual boards are used inefficiently.

G_j is recalculated after each board is processed, and is a moving average. This is an important characteristic of the control value in FLCO because of the assumption that board quality is not known before sawing. A moving control value provides two benefits. First, by regularly updating G_j, the FLC can adjust size values in response to any sustained rises or drops in board quality. Second, the FLC is guaranteed a goal that can be reached, enabling it always to make a useful control action. In contrast, consider an unrealistic goal, say that of filling a nonempty cutting order with zero boards. This goal would cause the FLC to increase all size values by some large amount, and the control action would have no effect in changing relative rates of recovery for sizes in the cutting order.

A_j, the average expected number of boards required to fill the cutting order is averaged with the value of G_{j-1} to prevent the recovered parts of any single board from changing the control value too abruptly. On one hand, nothing in the rough-mill environment exists to prevent a board from being followed by another of notably higher or lower quality. On the other hand, the quality of a particular board is not completely independent of the quality of the boards processed before or after it. With respect to their quality, boards are randomly, but not uniformly distributed (Brunner, 1996). Therefore including G_{j-1} in calculating G_j is important.

An FLC is commonly designed to make control decisions based on some values that measure error and change in error in a controlled system. Error describes how far a control parameter is from the goal, and change in error describes how fast it is approaching or departing from the goal. For example, if a control variable is close to the goal and approaching it very rapidly, the FLC uses this information to avoid overshooting the goal. FLCO measures error for some size i by e_{ij}-G_j, and measures change in error by $(e_{ij}-G_j) - (e_{ij-1}-G_{j-1})$.

So far, all of the parameters mentioned are crisp: each of them can be represented by some real number. Before applying fuzzy reasoning methods[4] to them, FLCO must "fuzzify," or convert to a corresponding fuzzy number, the crisp input parameters. This is the responsibility of the fuzzification interface.

---

[4] See Appendix 2 for a brief overview of fuzzy logic and fuzzy logic control.

4.2.3 Fuzzification interface

A fuzzifier maps a crisp point into a fuzzy set by a membership function (Mendel, 1995). In FLC applications, triangular membership functions are common because they are easy to understand and apply. Figure 5 shows the membership functions that FLCO uses to fuzzify error and change in error. These fuzzy sets conform to two design guidelines recommended by Cox (1992).

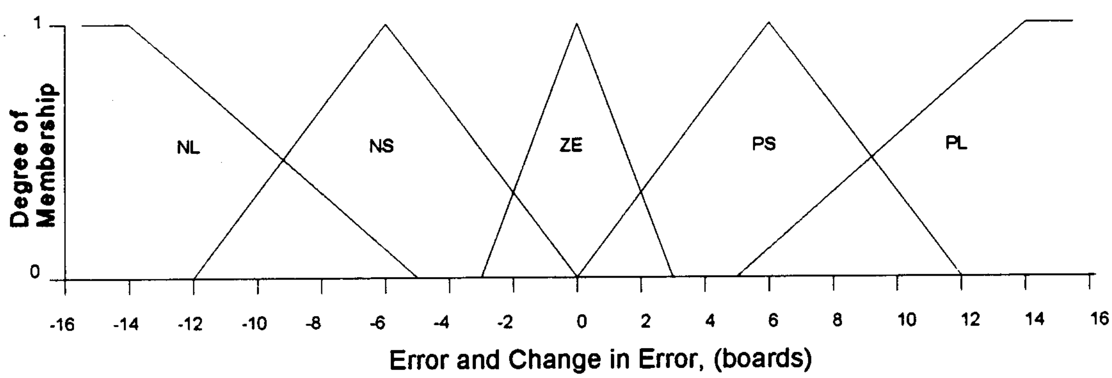

Figure 5 Fuzzy sets for error and change in error

First, the ideal value for error and change in error provides a center point for the fuzzy sets. Second, the density of the fuzzy sets decreases the further from the ideal control value they are, reflecting the decrease in certainty that the system is on target. Equation 3 defines the membership function for each of the fuzzy sets:

$$y = -|x - h|m + k \tag{3}$$

where

m absolute value of the slope of the triangle's sides;

h x-coordinate of the triangle's vertex;

k height of the triangle;

and is defined for all x such that $y \geq 0$. Each of the fuzzy sets has a name associated with it that gives a description of the set. Table 3 gives these names, their abbreviations and their values for m, h and k.

Table 3 Fuzzy set parameters for error and change in error

Name	Abbreviation	m	h	k
Negative Large	NL	1/9	-14	1
Negative Small	NS	1/6	-6	1
Zero	ZE	1/3	0	1
Positive Small	PS	1/6	6	1
Positive Large	PL	1/9	14	1

Exploratory tests which varied the values for m and h gave evidence that the values in Table 3 enabled the FLC to more effectively reduce the number of boards used to fill the cutting order than other values for m and h. The values in Table 3 can be interpreted as follows: if the expected number of boards needed to meet the demand for a given size is fourteen or more than the expected number of boards averaged over all sizes, that difference is large and an appropriate control action should be taken. Similarly, if the difference is about six, that difference is small and a different control action should be taken.

Fuzzy sets NL and PL are trapezoidal, not triangular. Any error or change in error less than or equal to -14 will have a membership in NL of 1.0. Similarly, any error or change in error greater than or equal to 14 will have a membership in PL of 1.0. This reflects the knowledge that any number less than -14 is definitely a large magnitude negative number, and any number greater than 14 is definitely a large magnitude positive number.

4.2.4 Decision making logic

Once the crisp measurements of error and change in error have been fuzzified, the fuzzy rules use them to determine a fuzzy change in value for the appropriate size. Lee (1990a) observes that a FLC "should always be able to infer a proper control action for every state of process," a condition he calls "completeness." For FLCO, this means that there should be a fuzzy rule for every combination of fuzzy error and fuzzy change in error. Since there are five possible fuzzy values for both error and change in error respectively, the rule base contains twenty-five rules. Figure 6 shows the fuzzy membership functions used for the consequents of each of the rules, and Table 4 gives the names and their values for m, h and k. In order to insure that the domain of these fuzzy sets is appropriate for any cutting order, FLCO normalizes the original size values to range between 100 and 1100. These values are unitless, and only serve to describe relative priorities between sizes. Like the fuzzy sets used for error and change in error, exploratory tests gave evidence that these values for m and h offer better performance than any others tested.

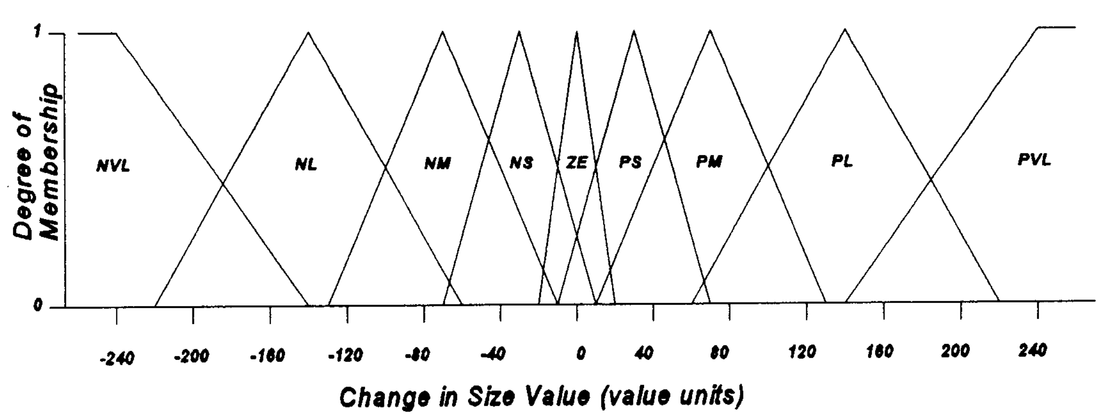

Figure 6 Fuzzy sets for change in size value

Table 4 Fuzzy set parameters for change in value

Name	Abbreviation	m	h	k
Negative Very Large	NVL	1/100	-240	1
Negative Large	NL	1/80	-140	1
Negative Medium	NM	1/60	-70	1
Negative Small	NS	1/40	-30	1
Zero	ZE	1/20	0	1
Positive Small	PS	1/40	30	1
Positive Medium	PM	1/60	70	1
Positive Large	PL	1/80	140	1
Positive Very Large	PVL	1/100	240	1

When the recovery rate of a size is high in relation to the average recovery rate, the error will be positive. When it is low, the error will be negative. So the fuzzy rules decrease the value of sizes with large positive errors, and increase the value of sizes with large negative errors. For large errors, the corresponding changes in value will be large, and for small errors, the corresponding changes in value will be small. In addition, the rules also consider change in error. By doing so, they reduce the chance that the system will overshoot the ideal control value, and reduce the time required to reach the control value. Table 5 shows the complete fuzzy rule matrix.

Table 5 Fuzzy rule matrix for change in value

Change in Error	Error NL	NS	ZE	PS	PL
NL	NVL	NL	NM	NS	ZE
NS	NL	NM	NS	ZE	PS
ZE	NM	NS	ZE	PS	PM
PS	NS	ZE	PS	PM	PL
PL	ZE	PS	PM	PL	PVL

As an example, suppose a size has an error the FLC labels "positive large", and a change in error also labelled "positive large". Table 5 shows that the change in value for that size is "positive very large." In other words, the size under consideration is overshooting the goal by a large amount (error), and what is more, the overshoot is increasing rapidly (change in error). This can happen when the size is not being recovered as fast as the others in the cutting order. To correct the situation, the FLC increases the size's value by a very large amount. An increase in the size's value changes its relative priority in the cutting order, resulting in a subsequent increase in recovery.

4.2.5 Defuzzification interface

For each size in the cutting order, FLCO finds a fuzzy set which describes the best control action to take. Having done so, it generates a crisp change in value for each size by taking the center of gravity of the fuzzy set. Once the crisp change in value is found, FLCO adds it to the corresponding size value. The new size values are then passed to CORY, which uses them to find the sawing solution for the next board. If the demand for a particular size is met, that size is dropped from the cutting order.

4.2.6 An example of FLC

Consider a simple cutting order with three sizes, shown in Table 6. Table 6 also shows the number of pieces of each size recovered and the current value of each size after 100 boards have been sawn. The following example illustrates the FLC process that takes place after the 101st board has been sawn.

Table 6 Example cutting order and system state

	Length x Width (in.): s_i	**Demand: d_i**	**Pieces recovered after 100 boards: $q_{i,100}$**	**Value of size after 100 boards**	**Pieces recovered from the 101st board: $q_{i,101} - q_{i,100}$**
Size 1	18.0 x 1.00	1000	99	1660	6
Size 2	33.0 x 2.25	250	29	1370	2
Size 3	75.0 x 2.00	500	46	8101	0

The rightmost column of Table 6 shows the number of pieces of each size recovered from the 101st board sawn. Recall from page 27 the variables the FLC uses to determine an objective value. For each size, the FLC calculates the expected number of boards required to meet demand:

$e_{1,101} = 1000 / (105 / 101) = 961.91.$

$e_{2,101} = 250 / (31 / 101) = 814.52$

$e_{3,101} = 500 / (46 / 101) = 1097.83.$

A_{101}, the average expected number of boards required to meet demand for all sizes after cutting 101 boards is 958.09. A_{101} is averaged with the objective value the FLC found after the 100th board: for illustrative purposes, suppose G_{100} equals 968.78. Then G_{101}, the weighted average expected number of boards required to meet demand for all sizes after cutting 101 boards equals

963.44. In other words after 101 boards are sawn, the FLC will aim to set size values so that the cutting order is filled when about 964 boards are sawn. For Size 1, the FLC finds a control action (ie. the amount to adjust its value) as follows. The error for Size 1 equals $e_{1,101}$ - G_{101}, or -1.53. The change in error for Size 1 equals ($e_{1,101}$ - G_{101}) - ($e_{1,100}$ - G_{100}), or -42.85. With these values, the FLC evaluates each rule in the rule base.

In this example, only two of the 25 rules have any effect. The top row of fuzzy sets in Figure 7 represent the fuzzy sets for the rule "if the error is Zero and the change in error is Negative Large, then the change in value is Negative Medium." The bottom row of fuzzy sets represents the rule "if the error is Negative Small and the change in error is Negative Large, then the change in value is Negative Large." In the top row, the error of -1.53 has a value in the fuzzy set ZE of 0.49, and the change in error of -42.85 has a value in the fuzzy set NL of 1.0. To find the firing strength of the rule, the FLC takes the minimum of these two values. The change-in-value fuzzy set is multiplied by 0.49 to obtain a result. A similar process occurs for the rule represented in the second row of Figure 7. The final fuzzy set from which a control value is obtained is the union of all fuzzy set results, represented by the rightmost fuzzy set in Figure 7.

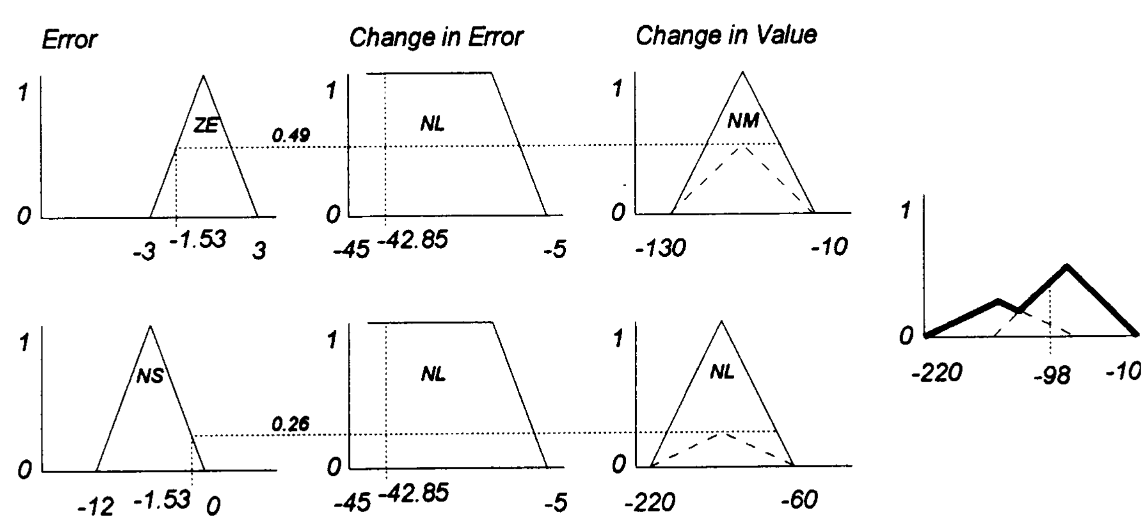

Figure 7 Example of fuzzy rule evaluation

The FLC finds the center of gravity of this fuzzy set which is about -98. Ninety-eight is subtracted from 1660 making the value of Size 1 after 100 boards are sawn 1562. This process of fuzzy inference and value adjustment is repeated for each of the sizes in the cutting order. After finding a change in value for each size, the FLC returns control to the FLCO software, which then executes CORY with the adjusted size values. CORY finds a sawing solution for the next board, and the entire process is repeated.

4.3 Software Implementation

FLCO is written in C++ and requires a Microsoft Windows 3.1 (or compatible) operating system (Microsoft, 1992). The CORY routines were originally written for a DOS operating system, but with only minor modifications have been incorporated into the Windows-based FLCO program.

FLCO allows the user to enter filenames for a cutting order, a board data file, and an output file. If no cutting order file exists, the user can create one by entering the length, width, demand and starting value for the sizes in the cutting order. FLCO enables the user to edit and save this list for future use. To create a board data file, the user can use any ASCII text editor. Three control strategies are available at run-time: FLC, drop sizes as demand is met, and cut to inventory. After the user has chosen one of these control strategies and run the controller, FLCO reports whether the cutting order was filled, and then provides a summary of the results. The results include number of boards processed, average percent area yield, average board processing time, and number of pieces recovered for each size in the cutting order.

Figure 8 shows the general flow of control in FLCO. After initialization and input, FLCO begins processing. While boards are available for cutting and while the cutting order is not yet filled, FLCO retrieves the data for the next available board and CORY finds a sawing solution for it. For each size in the cutting order, FLCO calculates how far the size deviates from the current objective value (error) and how fast it is approaching or departing from the current objective (change in error). With this information, FLCO can perform fuzzy logic control, calculating new values for CORY to use when sawing subsequent boards.

Figure 9 shows the general flow of control in the FLC. For every size in the cutting order, every rule in the rule base needs to be evaluated, requiring the nested loop structure shown in Figure 9. The outer loop represents the FLC's iteration through every size in the cutting order, and the inner loop represents the FLC's iteration through every rule in the rule base. The inner loop fuzzifies the crisp error and change in error, and finds a resultant fuzzy set for every rule in the rule base. In other words, every rule in the rule base is instantiated and fired. However, many rule firings will result in an empty fuzzy set, so they will not affect the change in value for the size under consideration. The union of the resultant fuzzy sets is returned to the outer loop. The outer loop finds a crisp change in value from these fuzzy sets and adjusts the value of the current size before considering the next size in the cutting order. This process is repeated until a change in value is found for every size.

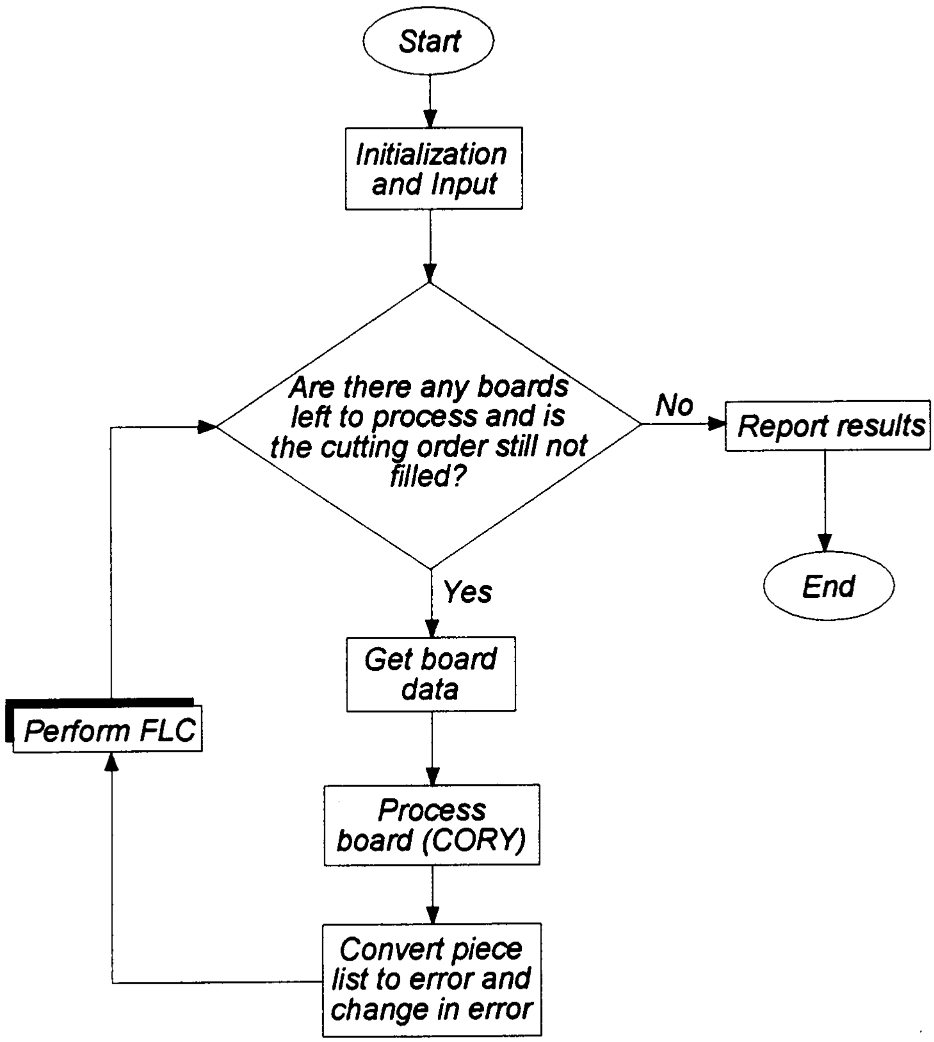

Figure 8 Flow chart for FLCO software

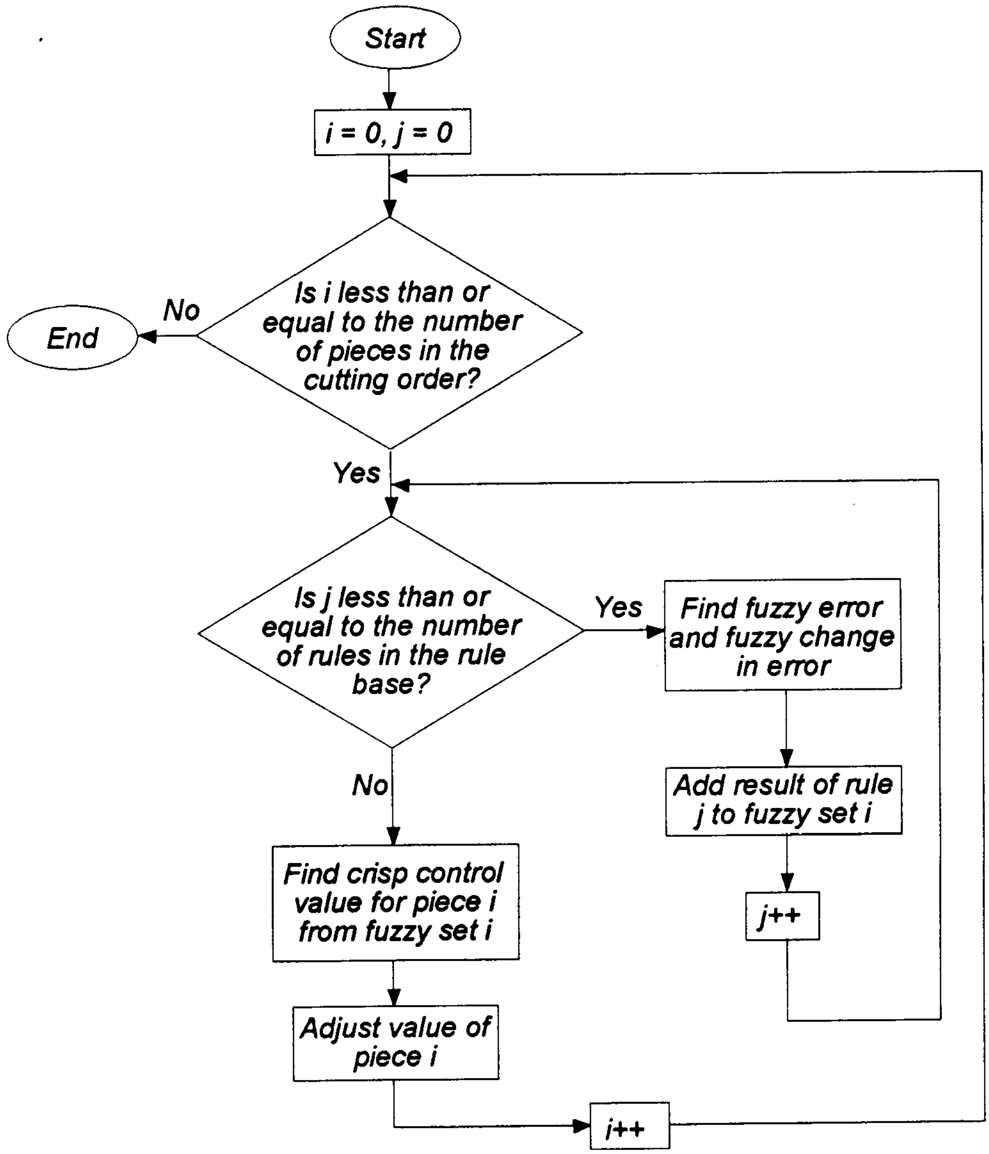

Figure 9 Flow chart for fuzzy logic controller portion of FLCO software

5. FLCO PERFORMANCE EVALUATION

Implemented according to the design described in the preceding chapter, the FLCO software meets the goals stated in Chapter 4. However, merely meeting those goals leaves at least two important questions unanswered. First, how does FLCO perform across a variety of lumber grades and cutting orders? Second, how well does it perform compared with other solution methods? To answer these questions, a series of tests was conducted which included three cutting bills, three lumber grades and two other solution methods.

5.1 Cutting Orders

These FLCO performance evaluations use three cutting bills, each of which is from a mill in the northwestern United States. The demands are randomly associated with each size, and range between 10 pieces and 110 pieces, in increments of 20. This range was chosen because of the number of boards in the data sets: a higher range could result in a cutting order that is impossible to fill with the available boards. Table 7 briefly summarizes each of the cutting orders. Size values are unitless; they represent relative size priorities.

Table 7 Cutting order summary

		Length Range	Width Range	Value Range
Cutting Order 1	minimum	10.5 in.	1.925 in.	4
(40 sizes)	maximum	65.0 in.	5.562 in.	314
Cutting Order 2	minimum	10.562 in.	1.875 in.	162
(50 sizes)	maximum	84.062 in.	5.625 in.	3897
Cutting Order 3	minimum	25.0 in.	2.125 in.	90
(26 sizes)	maximum	84.0 in.	4.875 in.	853

Cutting Order 2 originally contained fifty-one sizes, but the largest size, 84.062 inches by 4.875 inches, was removed because its demand could not be met with the lowest grade of boards. Similarly, Cutting Order 3 originally contained thirty sizes, but the four largest sizes were removed because their demands could not be met with the lowest grade boards. In a rough mill, these exceptionally large pieces are usually not treated as the other pieces in the cutting order. Because they are so valuable and so rare, a ripsaw operator will sacrifice recovery of every other piece to recover one of these large ones. To consider this exceptional condition, FLCO would need to contain program logic separate from the FLC. By removing these pieces from the cutting orders, sawing decisions are based on values assigned solely by the FLC, and not by any extraneous logic.

5.2 Board Data

Molding and millwork manufacturers in the western United States commonly use Ponderosa Pine as their stock material. Because the cutting orders described in the previous section are from mills in this region, these

FLCO performance evaluations use Ponderosa Pine data as input. The United States Department of Agriculture's Forest Products Laboratory in Madison, Wisconsin provided the original board data. McDonald et al. (1981) used these same data to develop 5/4 Ponderosa Pine yield tables, and they include three lumber grades: No. 1, No. 2, and No. 3 Shop. Table 8 shows the number of boards and number of board feet in each grade.

Table 8 5/4 ponderosa pine shop sample information

Grade	Number of Boards	Board Feet
No. 1	386	8534
No. 2	1537	26797
No. 3	1361	17985

The original data records board information in 0.2500 inch Cartesian coordinate units, while the CORY libraries FLCO uses expect board data to be in 0.0001 inch units. Like Rahardjo (1992), the board data was transformed using a pseudo-random number generator and a uniform distribution function, in this case to create board and defect coordinates of 0.0001 inches. This procedure assumes that board and defect dimensions are uniformly distributed within their original 0.2500 inch coordinates. This is a reasonable assumption because defect dimensions are continuous and the original measurements were inclusive of defects (McDonald et al., 1981).

5.3 Other Solution Methods

Besides fuzzy logic control, two other heuristic control methods were examined. One of these is a very simple approach, and provides a sort of baseline comparison, while the other is more complicated.

5.3.1 Dropping sizes from the cutting order

This first approach is simply to drop sizes from the cutting order as their demands are met. Once the demand for a size is filled, it will not be considered for cutting in any subsequent boards. The absolute value for each size in the cutting order remains constant throughout processing. However, the relative value for each size will change as pieces are removed from the cutting order. For example, consider a cutting order with three sizes, A, B and C, with values 5, 10 and 20, respectively. At the start of processing, the value of size A is only 25 percent that of the largest size in the cutting order. Suppose that some time during processing, the demand for size C is met, and it is dropped from the cutting order. Now the value of size A is 50 percent that of the largest size in the cutting order. This is a significant increase in the relative priority of size A, and CORY would correspondingly attempt to recover more pieces of size A than it had previously. This drop sizes approach can be chosen at runtime by the user.

5.3.2 Complex dynamic exponential prioritization

Thomas (1995a) presents a prioritization function that he calls a Complex Dynamic Exponential (CDE) strategy. Its dynamic qualities lie in its responsiveness to the demand for a cutting size, and to the number of pieces of that size recovered. As boards are processed, piece values are assigned according to Equation 5, with values for WF_{Length} and WF_{Width} assigned by Equation 4.

$$WF = \sqrt{\ln(demand * \max(1, (35 - count)) * 15)} * MF + 1.0 \tag{4}$$

$$Value = Length^{WF_{Length}} * Width^{WF_{Width}} \tag{5}$$

For WF_{Length}, MF = 0.14, and for WF_{Width}, MF = 0.07. Count represents the number of pieces recovered of the size for which WF is being calculated.

By design, FLCO can easily accommodate any method of assigning values to pieces. For these tests, the CDE formulas are substituted for the FLC routines, and the FLCO program is recompiled.

6. RESULTS AND DISCUSSION

Tables 9, 10 and 11 show the results of running FLCO with the various cutting orders, lumber grades and control strategies described in the previous chapter. When using the strategy of dropping sizes from the cutting order, FLCO could not fill Cutting Order 3 with the available No. 3 Shop board data. Therefore Tables 9, 10 and 11 do not contain results for this combination of cutting order and lumber grade.

Table 9 FLC performance results (number of boards, average percent area yield)

	Cutting Order 1	Cutting Order 2	Cutting Order 3
No. 1 Shop	161 61.61	203 60.27	202 51.57
No. 2 Shop	227 50.66	262 51.10	289 39.33
No. 3 Shop	373 35.40	404 38.66	

Similar patterns exist in each table. First, each control strategy uses fewer high grade boards to fill a given cutting order than low grade boards. Second, each strategy fills Cutting Order 1 with fewer boards than the other two cutting orders for all lumber grades. The single exception to this is the drop sizes strategy, which used eleven more boards to fill Cutting Order 1 than Cutting Order 2 when sawing No. 3 Shop lumber. Third, the CDE strategy produces higher yields than FLC, and FLC produces higher yields than dropping sizes. One reason for CDE's high yields is that it will continue to recover pieces of a size even after the demand for that size has been met.

Table 10 Drop sizes performance results (number of boards, average percent area yield)

	Cutting Order 1	Cutting Order 2	Cutting Order 3
No. 1 Shop	178 56.31	199 58.80	203 50.62
No. 2 Shop	253 46.19	276 48.69	302 38.56
No. 3 Shop	456 29.31	445 34.88	

Table 11 Complex dynamic exponential prioritization performance results (number of boards, average percent area yield)

	Cutting Order 1	Cutting Order 2	Cutting Order 3
No. 1 Shop	159 64.40	213 63.68	200 54.10
No. 2 Shop	232 57.42	270 57.08	285 42.44
No. 3 Shop	378 48.42	417 48.24	

Before comparing FLC's performance with other control strategies and before examining its effects across different cutting orders and lumber grades, it is interesting to examine some behaviors of the FLC to find out if it is behaving as it was designed. In particular, by examining error and change in error for each size--the variables on which the FLC makes control decisions--one can see whether the FLC is properly controlling piece recovery.

6.1 Piece Error and Change in Error

Rather than examining error and change in error for every combination of size, cutting order and lumber grade, three typical examples are shown here. These examples are from FLCO processing of No. 1 Shop lumber to fill Cutting Order 1, and represent a variety of sizes and demands.

Figure 10 shows the expected number of boards required to recover ninety 13.5 x 1.925 inch pieces. This size is one of the smallest in the cutting order, and has relatively large demand. While processing the first twenty or thirty boards, the average expected number of boards is in a transient period. During this transient period the expected number of boards required to meet the

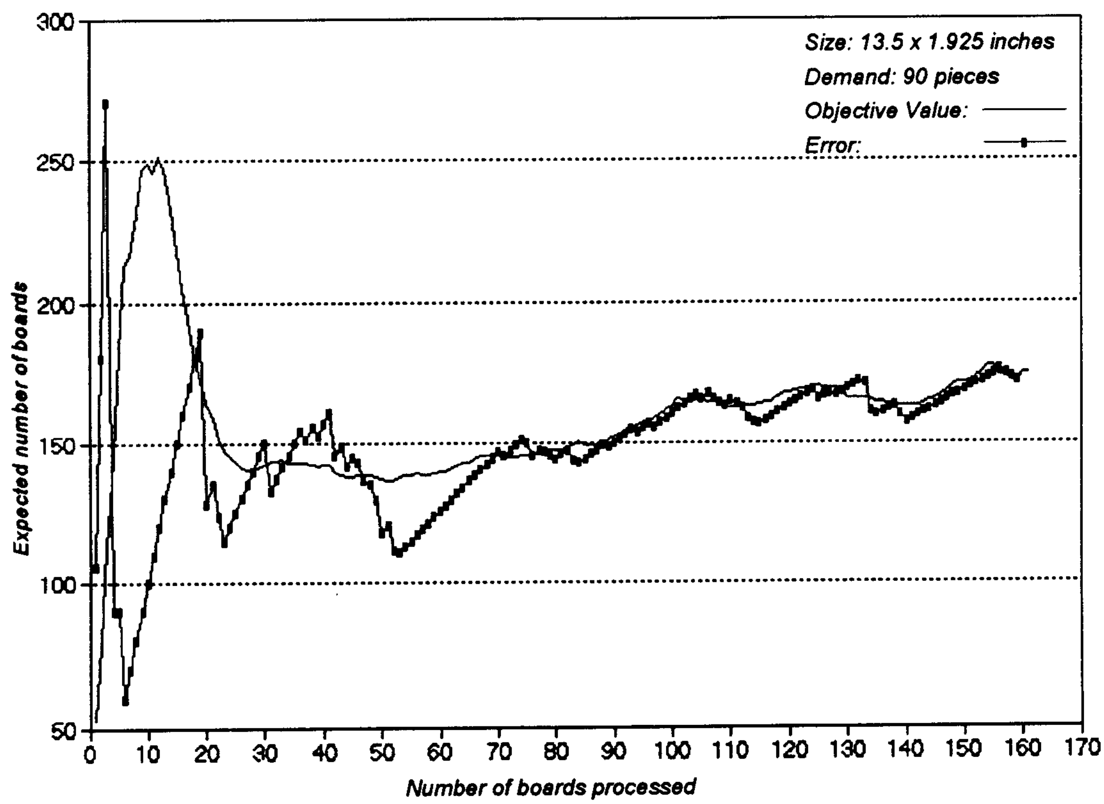

Figure 10 Expected number of boards to recover ninety 13.5 x 1.925 inch pieces

demand for this size is large, alternately larger and smaller than the objective value. Between boards forty and fifty-five, the decreasing trend in error seems to reflect an overcompensation by the FLC. A run of positive error preceding this causes the FLC to increase the size's value repeatedly. Then, when a change in board quality occurs allowing for greater recovery of that size, the error drops sharply. From about board seventy onward, the FLC maintains control over this size's recovery, keeping it close to the objective.

Figure 11 shows the expected number of boards required to meet demand for ten 51.0 x 2.425 inch pieces, one of the larger sizes in the cutting order. Because its demand is very small, whenever a piece of this size is recovered, the error decreases sharply. The FLC attempts to compensate by

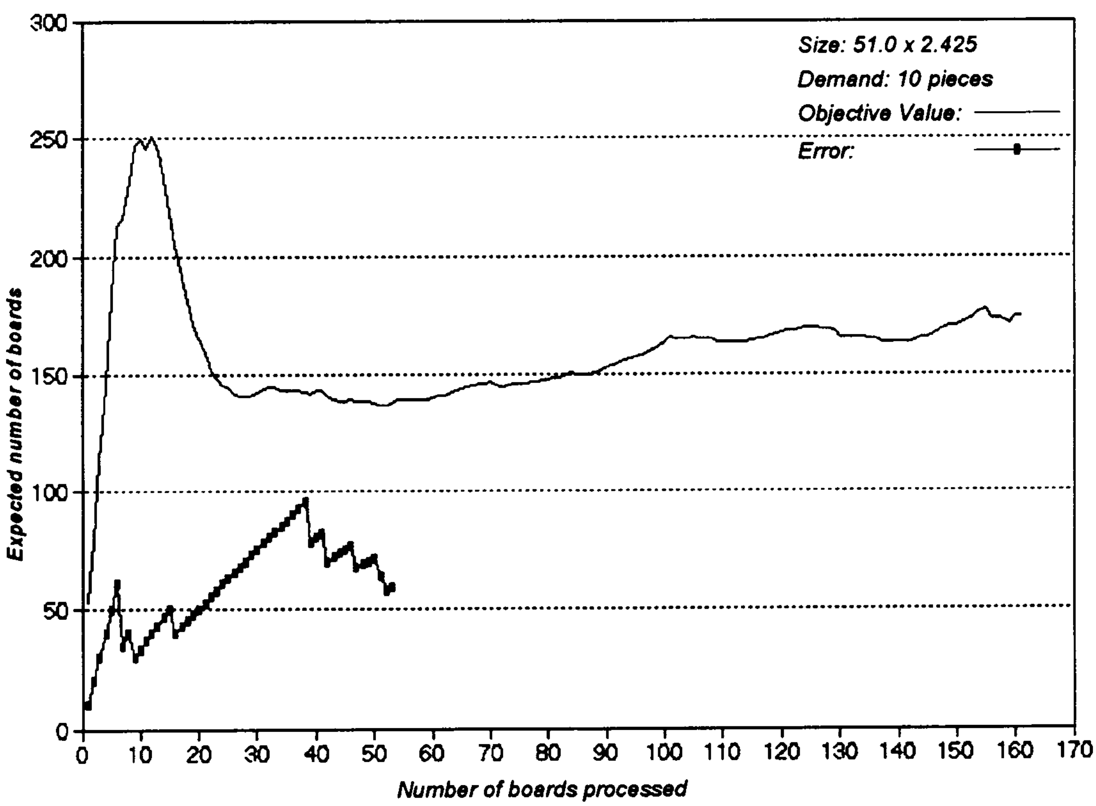

Figure 11 Expected number of boards to recover ten 51.0 x 2.425 inch pieces

lowering the value of the size, but because the FLC does not assign values less than one, whenever a clear area can contain a piece of just this size, CORY will recover it. This happens within the first fifty-three boards. The demand for the size is met and it is dropped from the cutting order.

Figure 11 also shows a clear upward trend of the expected number of boards required to fill the cutting order beginning around board sixty. One possible explanation for this is that the FLC makes changes in size values that cause CORY to make poor sawing decisions. However, this does not seem to be the case. The average percent area yield for the first sixty boards is 62.49, while the average percent area yield for the remaining boards is 61.08, a difference slightly less than 1.5 percent. Another possible explanation is that a decrease in board quality begins around board sixty. The average clear area size for the first sixty boards is 291.39 square inches, while the average clear area size for the remaining boards is 270.51 square inches. This difference of 20.88 square inches is about 21 percent of the average size in the cutting order, and is most likely the cause of the increase in the objective value.

As a final example, Figure 12 shows the expected number of boards required to recover thirty 63.0 x 3.425 inch pieces. During the objective value's transient period, the error for this size becomes very large. However, the FLC compensates and adjusts the value of the size so that its rate of recovery is increased. Notice the "saw-tooth" behavior of the error, also seen in Figure 11: sharp decreases in error followed by slow increases. Recall that error is defined by

$$\frac{\textit{Demand}}{\textit{Rate}} \tag{6}$$

and rate in turn is defined by

$$\frac{\textit{Number of Pieces Recovered}}{\textit{Number of Boards Processed}}. \tag{7}$$

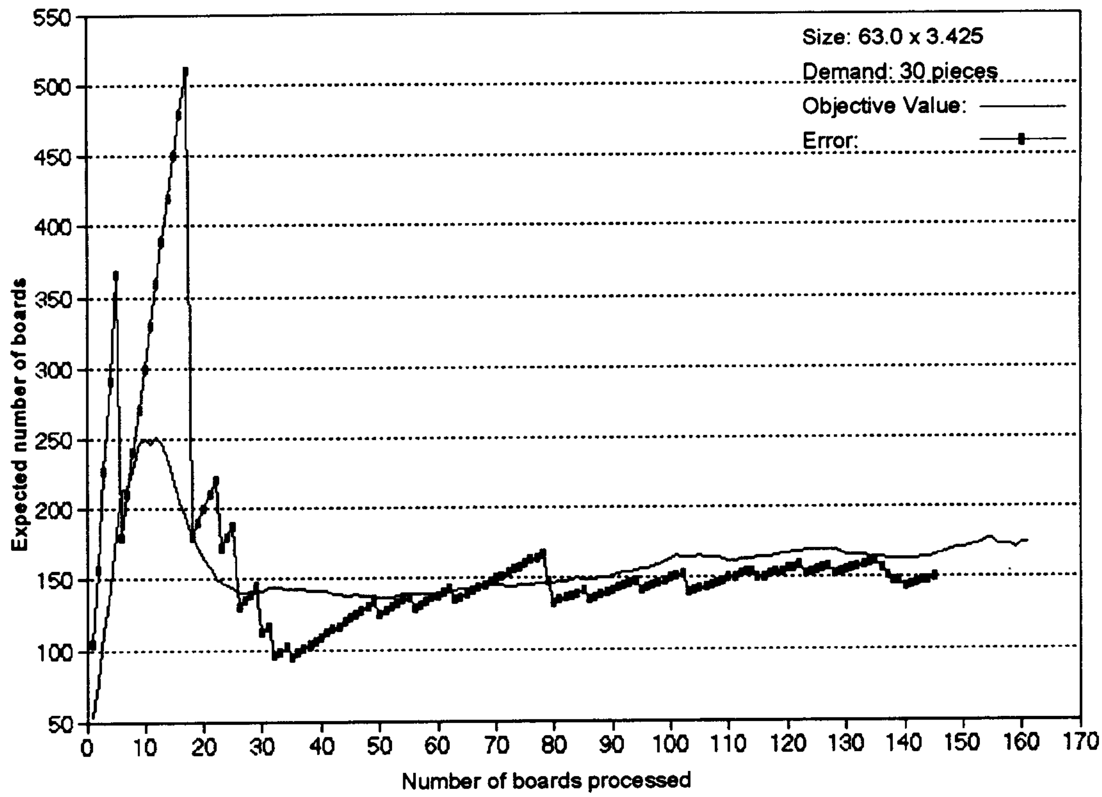

Figure 12 Expected number of boards to recover thirty 63.0 x 3.425 inch pieces

When the number of boards processed is small, a unit of change in the numerator of Equation 7 makes a greater difference than a unit of change in the denominator. So Figure 12 shows a sharp decrease when even a single piece is recovered, while the subsequent slow increase is due to changes in the number of boards processed.

6.2 Effect of FLC Across Lumber Grades

As indicated in the previous chapter, one of the simplest heuristic approaches to filling a cutting order is to drop sizes from the order as their demands are met. While this strategy is probably never used by itself in a mill, it provides a baseline by which to compare the behavior of the FLC. From Tables 9 and 10, one can find the difference in number of boards required to fill the cutting order between the FLC and drop sizes control strategies. For all lumber grades and all cutting orders, the observed mean percent difference in the number of boards used between FLC and dropping sizes is 6.89. The two-sided p-value of 0.0181 indicates that there is a significant difference between the two control strategies.

Averaged over the three cutting orders, FLCO met demands with 2.68 percent fewer No. 1 Shop boards using FLC as a control strategy than by dropping sizes as a control strategy. Similarly, using FLC requires an average of 6.55 percent and 13.71 percent fewer No. 2 Shop and No. 3 Shop boards, respectively, than dropping sizes does. This implies that FLC is more effective for low grade boards than high grade boards. By definition, high grade boards will have more large clear areas than low grade boards, providing more opportunities to recover pieces regardless of their size. Thus the consequences of poor sawing decisions are not severe when cutting high grade lumber. However, when cutting low grade material, judicious use of the few large clear areas that are available becomes more important. FLC enables CORY to make good sawing decisions for improved use of low grade boards.

Even though the reductions observed for the No. 1 Shop boards are smaller than those for No. 2 and No. 3 Shop boards, they may have economic significance. On average, high grade lumber has a greater monetary value per board than low grade lumber. This implies that the monetary savings realized from using one less No. 1 Shop board is greater than the savings realized from using one less No. 3 Shop board.

6.3 Effect of FLC Across Cutting Order

Averaged over the three lumber grades, FLCO filled Cutting Order 1 with 12.68 percent fewer boards using FLC as a control strategy than by dropping sizes as a control strategy. Similarly, using FLC requires an average of 4.09 percent and 2.40 percent fewer boards than dropping sizes for Cutting Orders 2 and 3, respectively. Based on the number of large sizes it contains, Cutting Order 3 appears to be the most "difficult" of the three orders to fill. Together these observations seem to imply that FLC is more effective when filling cutting orders that are relatively easy to satisfy than when filling difficult cutting orders. Some additional tests provide further support that this is the case.

Cutting Order 1 was modified so that two sizes, 23.0 x 2.237 inches and 23.0 x 2.425 inches, have demands of 500 pieces each. Because of the increased demands, this modified cutting order should be more difficult to fill than the original Cutting Order 1. Averaged over the three lumber grades, FLCO filled the modified cutting order with 8.34 percent fewer boards by using FLC as a control strategy than by dropping boards. This percentage is smaller than the 12.68 percent observed when processing the unmodified Cutting Order 1, suggesting that a more difficult cutting order decreases the effectiveness of FLC. As the next section explains, the CDE prioritization strategy does not have this problem, so it is likely that the FLC is simply not responsive enough to handle difficult cutting orders as effectively as simpler cutting orders.

6.4 FLC Versus Complex Dynamic Exponential Prioritization

From Tables 11 and 10, one can find the difference in number of boards required to fill the cutting order between the CDE and drop sizes control strategies. Averaged over the three lumber grades, FLCO filled Cutting Order 1 with 12.03 percent fewer boards using CDE as a control strategy than by dropping sizes as a control strategy. Similarly, using CDE requires an average of 1.43 percent and 3.55 percent fewer boards than dropping sizes for Cutting Orders 2 and 3, respectively.

Comparing these percentages with those resulting from FLC, it appears that FLC is only slightly more effective than CDE control for Cutting Order 1 and about 2.5 percent more effective for Cutting Order 2. For Cutting Order 3, the most difficult cutting order, FLC is about 1 percent less effective than CDE. Because FLC does not set size values but only increases or decreases them, some "rise time" is always present when the system behavior is approaching, but does not match, the desired behavior. Figures 10, 11 and 12 show this. CDE on the other hand sets size values directly, and can more quickly control the behavior of the system. During the rise time, FLC may miss opportunities to recover some sizes in the cutting order. When a cutting order contains sizes that are difficult to obtain, more boards need to be processed to make up for the lost opportunities.

Averaged over the three cutting orders, FLCO used 1.71 percent fewer No. 1 Shop boards using CDE as a control strategy than by dropping sizes as a control strategy. Similarly, CDE used an average of 5.37 percent fewer No. 2 Shop and 11.70 percent fewer No. 3 Shop boards than dropping sizes for Cutting Orders 2 and 3, respectively. Compared with the difference in number of boards between FLC and dropping sizes, these percentages are about 1 percent less for both No. 1 Shop and No. 2 Shop boards, and about 2 percent less for No. 3 Shop boards. These percentages suggest that FLC is slightly more effective than CDE at reducing the number of boards required to meet a cutting order for a variety of grades.

While these results are suggestive of a difference between the two control strategies, that conclusion is not supported by statistical analysis. For all lumber grades and all cutting orders, the observed mean percent difference in the number of boards used between FLC and CDE is 1.32. The two-sided p-value of 0.1494 indicates that there is not a significant percent difference between the two groups.

One difference between the two strategies is the source of the starting values for sizes in a cutting order: CDE generates its own, while FLC accepts user-supplied values. User-supplied values may be poorly chosen, requiring FLC to cut more boards than if the values were well chosen. To test whether the starting values made a difference between the two control strategies, the cutting orders were modified so that the starting size values were equal to the size values generated by CDE. The three modified cutting orders were then filled with boards from the three lumber grades as before. The statistical analysis shows that the mean percent difference in the number of boards used to fill the modified cutting orders and the original cutting orders is 0.14, with a two-sided p-value of 0.9469. This suggests that starting values do not affect the performance of FLC relative to CDE.

7. CONCLUSIONS

In an attempt to reduce the number of boards needed to fill cutting orders in the secondary manufacture of lumber, FLCO offers one possible approach. Embodying the ideas of FLC, it considerably reduced the number of boards required compared with a very simple control method, and showed little difference in reducing the number of boards required compared with a more complicated control method.

FLCO incorporates CORY lumber cut-up software, and offers an analytical tool to study the effects of various control methods on lumber use. The C++ code in which FLCO is written has a highly modular structure allowing for easy exchange of coded control methods. In addition, the interface between FLCO and CORY is very small, allowing a different version of CORY to be used or even a different lumber cut-up software package altogether.

Three cutting orders from actual rough mills and data sets containing boards representing three grades of lumber provided the test data for the FLC. The FLC behaves as it was designed to, properly adjusting the values of cutting sizes to reduce the number of boards required to fill a cutting order. Two other control methods--dropping sizes and CDE prioritization--provided external standards by which FLC was compared. Overall, FLC considerably reduces the number of boards required to meet a cutting order when the cutting order is not very difficult to fill, or when the lumber quality is low. FLC offers smaller reductions for difficult cutting orders or when the lumber quality is high. CDE prioritization offers reductions similar to those of FLC.

FLC and CDE gave large reductions in two cases: when cutting low grade boards, and when filling cutting orders containing no exceptionally large sizes and no exceptionally large demands. This indicates the importance of using some form of intelligent control when cutting low grade lumber and when filling simple cutting orders. The smaller reductions observed when cutting high grade lumber may be important because of its high cost. Further study is needed to determine whether these small reductions have an economic benefit. Because FLC and CDE offer such similar results, it is not clear if one of the two control methods is more desirable to use than the other.

A solution to the diminished effects of FLC for difficult cutting orders may lie in further refinement of the fuzzy sets. A more thorough exploration of fuzzy sets over a wider variety of cutting orders may improve the performance of the FLC. Also, the performance might be improved if the knowledge base contained different fuzzy sets depending on the difficulty of the cutting order. Fuzzy sets that allow for larger changes in size values would enable the FLC to bring sizes to their desired values more quickly.

In addition, more work is required to refine the realism of the system FLCO models. Currently FLCO assumes that sizes will not be added to the cutting order during processing. In reality, a mill manager may try to fill cutting orders from several customers at a time. Rather than fill one customer's order before beginning to cut for another order, a mill manager would likely begin cutting the second customer's order before completion of the first customer's order. The FLC was not designed to consider this situation and more work is required to find out how its performance would be affected by sizes added to the cutting order during processing.

BIBLIOGRAPHY

Adamowicz, A., and A. Albano. 1976. A solution of the rectangular cutting stock problem. IEEE Transactions on Systems, Man, and Cybernetics SMC-6(4):302-310.

Albano, A., and R. Orsini. 1980. A heuristic solution of the rectangular cutting stock problem. The Computer Journal 23(4):338-343.

Anderson, J.D., C.C. Brunner, and A.G. Maristany. 1992. Effect of sawing stages on fixed-width, fixed-dimension yields. Forest Products Journal 42(11/12):74-78.

Azarm, S., G. Harhalakis, M. Srinivasan, and P. Statton Jr. 1991. Heuristic Optimization of rough-mill yield with production priorities. Journal of Engineering for Industry. 113(1):108-116.

Brunner, C.C. 1984. CORY -- a computer program to determine furniture cutting yields from both rip-first and crosscut-first sawing sequences. Ph.D. dissertation, Virginia Tech., Blacksburg, VA.

Brunner, C.C. 1996. Personal communication, February 29. Oregon State University, Corvallis, OR.

Brunner, C.C., and J.D. Anderson. 1991. Comparison of exhaustive- and heuristic-based versions of the CORY lumber cut-up program. Presentation at the 45th Forest Products Research Society Annual Meeting. Oregon State University, Corvallis, OR.

Brunner, C.C., D.A. Butler, A.G. Maristany, and D. VanLeeuwen. 1990. Optimal clear-area sawing patterns for furniture and millwork blanks. Forest Products Journal 40(3):51-56.

Brunner, C.C., M.S. White, F.E. Lamb, and J.G. Schroeder. 1989. CORY: a computer program for determining dimension stock yields. Forest Products Journal 39(2):23-24.

Carino, H.F. 1986. An input/output model for material requirements planning and cost allocation in wood products assembly. Forest Products Journal, 36(11/12):51-55.

Carino, H.F., and S.U. Foronda. 1990. SELECT: a model for minimizing blank costs in hardwood furniture manufacturing. Forest Products Journal 40(5):21-26.

Carino, H.F., and C.H. LeNoir, Jr. 1988. Optimizing wood procurement in cabinet manufacturing. Interfaces 18(2):10-19.

Carnieri, C., G.A. Mendoza, and L. Gavinho. 1994. Optimal cutting of lumber and particleboards into dimension parts: some algorithms and solution procedures. Wood and Fiber Science, 26(1):131-141.

Carnieri, C., G.A. Mendoza, and W.G. Luppold. 1993. Optimal cutting of dimension parts from lumber with a defect: a heuristic solution procedure. Forest Products Journal, 43(9):66-72.

Chambers, M.L. and R.G. Dyson. The cutting stock problem in the flat glass industry -- selection of stock sizes. Operational Research Quarterly 27(4):949-957.

Cheng, R.M.H., A.E. Fahim, and R. DiMichele. 1977. A computer-aided wood cutting system for maximum utilization of material. ISA Adv. Instrum., 32(1):65-73.

Cheng, R.M.H., and A.W. Pila. 1977. Maximized utilization of surface areas with defects by the dynamic programming approach. ISA Adv. Instrum., 32(1):51-64.

Christofides, N., and C. Whitlock. 1977. An algorithm for two-dimensional cutting problems. Operations Research 25(1):30-44.

Cox, E. 1992. Fuzzy fundamentals. IEEE Spectrum 29:59-61.

DeVor, R.E., T.H. Chang, and J.W. Sutherland. 1992. Statistical quality design and control: contemporary concepts and methods. Macmillan Publishing Company, New York, NY.

Dmitrovic, B., D. Wolf, and R.H.C. Husson. 1992. A standard model of the placement problem aimed at resolving a cutting stock problem. IEEE Transactions on Industry Applications 28(6):1406-1415.

Dubois, D., H. Prade. 1980. Fuzzy sets and systems: theory and applications. Academic Press, New York.

Dyckhoff, H. 1990. A typology of cutting and packing problems. European Journal of Operational Research 44:145-159.

Eisemann, K. 1957. The trim problem. Management Science 3(3):279-284.

Farley, A.A. 1990. Selection of stockplate characteristics and cutting style for two dimensional cutting stock situations. European Journal of Operational Research 44:239-246.

Foronda, S.U., and H.F. Carino. 1991. A heuristic approach to the lumber allocation problem in hardwood dimension and furniture manufacturing. European Journal of Operational Research 54:151-162.

Giese, P.J., and J.D. Danielson. 1983. CROMAX -- a crosscut-first computer simulation program to determine cutting yield. General Technical Report FPL 38. USDA Forest Service, Forest Products Laboratory, Madison, WI.

Giese, P.J., and K.A. McDonald. 1982. OPTYLD -- a multiple rip-first computer program to maximize cutting yields. USDA Forest Service Research Paper FPL 412. Forest Products Laboratory, Madison, WI.

Gilmore, P.C., and R.E. Gomory. 1961. A linear programming approach to the cutting stock problem. Operations Research 9(6):849-859.

Gilmore, P.C., and R.E. Gomory. 1963. A linear programming approach to the cutting stock problem - part II. Operations Research 11(6):863-888.

Gilmore, P.C., and R.E. Gomory. 1965. Multistage cutting stock problems of two and more dimensions. Operations Research 13(1):94-120.

Gilmore, P.C., and R.E. Gomory. 1966. The theory and computation of knapsack functions. Operations Research 14(6):1045-1074.

Hafley, W.L., and K.W. Hanson. 1973. Optimum sequence of cutting bills. Forest Products Journal 23(8):26-29.

Hahn, S.G. 1968. On the optimal cutting of defective sheets. Operations Research 16(6):1100-1114.

Hallock, H., and P. Giese. 1980. Does gang ripping hold the potential for higher clear cutting yields. USDA Forest Service Research Paper FPL 369. Forest Products Laboratory, Madison, WI.

Hamilton, E. 1996. Personal communication, April 10. Oregon State University, Corvallis, OR.

Herz, J.C. 1972. Recursive computational procedure for two-dimensional stock cutting. IBM Journal of Research and Development 16:462-469.

Hoff, K.G., E.L. Adams, and E.S. Walker. 1991. GR-1ST: PC program for evaluating gang-rip first board cut-up procedures. General Technical Report NE 150. USDA Forest Service, Northeastern Forest Experiment Station, Radnor, PA.

Kantorovich, L.V. 1960. Mathematical methods of organizing and planning production. Management Science 6(4):366-422.

Kantrowitz, M., E. Horstkotte, and C. Joslyn. 1995. Answers to frequently asked questions about fuzzy logic and fuzzy expert systems. comp.ai.fuzzy, ftp.cs.cmu.edu:/user/ai/pubs/faqs/fuzzy/fuzzy.faq, mkant+fuzzy-faq@cs.cmu.edu.

Lee, C.C. 1990a. Fuzzy logic in control systems: fuzzy logic controller -- part I. IEEE Transactions on Systems, Man, and Cybernetics 20(2):404-418.

Lee, C.C. 1990b. Fuzzy logic in control systems: fuzzy logic controller -- part II. IEEE Transactions on Systems, Man, and Cybernetics 20(2):418-435.

Mamdani, E.H. 1974. Applications of fuzzy algorithms for control of a simple dynamic plant. Proc. IEE 121(12):1585-1588.

Martens, D.G., and R.L. Nevel Jr. 1985. OPTIGRAMI: optimum lumber grade mix program for hardwood dimension parts. Research Paper NE 563. USDA Forest Service, Northeastern Forest Experiment Station, Radnor, PA.

McDonald, K.A., P.J. Giese, and R.O Woodfin. 1981. 5/4 ponderosa pine shop grade cutting yields. USDA Forest Service Research Paper FPL 394. Forest Products Laboratory, Madison, WI.

Mendel, J.M. 1995. Fuzzy logic systems for engineering: a tutorial. Proceedings of the IEEE 83(3):345-377.

Microsoft Corp. 1992. Microsoft Windows 3.1 user's guide. Microsoft Corp., Redmond, WA.

NHLA. 1986. Rules for the measurement and inspection of hardwood and cypress. National Hardwood Lumber Association, Memphis, TN.

Pegels, C.C. 1967. Heuristic scheduling models for variants of the two-dimensional cutting-stock problem. Tappi 50(11):532-535.

Rahardjo, S. 1992. The effect of rip-kerf width on 5/4 Ponderosa Pine cutting stock yield. M.S. thesis. Oregon State University, Corvallis, OR.

Rönnqvist, M. 1995. A method for the cutting stock problem with different qualities. European Journal of Operational Research 83:57-68.

Scheithauer, G., and J. Terno. 1988. Guillotine cutting of defective boards. Optimization 19(1):111-121.

Schott, K.A. 1995. The effect of edging red alder lumber on cut stock production. M.S. thesis. Oregon State University, Corvallis, OR.

Stern, A.R., and K.A. McDonald. 1978. Computer optimization of cutting yield from multiple-ripped boards. USDA Forest Service Research Paper FPL 318. Forest Products Laboratory, Madison, WI.

Sugeno, M. 1985. An introductory survey of fuzzy control. Information Sciences 36:59-83.

Sweeney, P.E., and E.R. Paternoster. 1991. Cutting and packing problems: an updated literature review. Working Paper No. 654, University of Michigan School of Business Administration. Ann Arbor, MI.

Thomas, R.E. 1995a. Prioritizing parts from cutting bills when gang-ripping first. USDA Forest Service, Northeastern Forest Experiment Station, Radnor, PA. (Unpublished).

Thomas, R.E. 1995b. ROMI-RIP: rough mill rip-first simulator. General Technical Report NE 206. USDA Forest Service, Northeastern Forest Experiment Station, Radnor, PA.

Thomas, R.E., C.J. Gatchell, and E.S. Walker. 1994. User's guide to AGARIS: advanced gang-rip simulator. General Technical Report NE 192. USDA Forest Service, Northeastern Forest Experiment Station, Radnor, PA.

Thomas, R. J. 1962. The rough-end yield research program. Forest Products Journal 12(11):536-537.

Voigt, J.V. 1987. Solving cutting stock problems of the furniture industry via man-machine communication. Wissenschaftliche Zeitschrift der Technischen Hochschule Ilmenau 33(6):113-118.

WWPA. 1988. Western lumber grading rules 88. Western Wood Products Association, Portland, OR.

Wodzinski, C., and E. Hahm. 1966. A computer program to determine yields of lumber. USDA Forest Service, Forest Products Laboratory, Madison, WI.

Zadeh, L. 1965. Fuzzy sets. Information and Control 8:338-353.

APPENDICES

Appendix 1 - Glossary of Selected Terms

cutting

a rectangular portion of a board obtained by a series of crosscut (across the width of a board) and rip (along the length of a board) sawing operations; a cutting is distinguished from waste in that its dimensions and features are desired and/or allowed by some customer.

cutting bill

a list of piece sizes and a value for each.

cutting order

a cutting bill in which each size has an associated demand.

demand

the number of pieces that need to be obtained from the input boards to satisfy a customer's requirements.

FLC

acronym for "fuzzy logic control" or "fuzzy logic controller," depending on the context.

FLCO

acronym for "fuzzy logic control/cutting order," the name of the software developed for this thesis.

order

see "cutting order."

piece

a cutting from a board; described in terms of its size.

rough mill

the area of fabrication that involves cutting boards into rough size lengths and widths (Azarm et al., 1991)

size

the length and width of a piece.

Appendix 2 - A Brief Overview of Fuzzy Logic

Fuzzy Logic was originally developed by Lotfi Zadeh in the 1960's as a means to capture the vagueness inherent in natural language (Kantrowitz, 1995; Zadeh, 1965). Mamdani (1974) did the first research into fuzzy logic control, based on Zadeh's prior work. Since then, Zadeh's original ideas have been developed further (eg. Dubois and Prade, 1980), and the concepts of Fuzzy Logic Control (FLC) have been the subject of far too many publications to mention. Rather than explaining ideas explained elsewhere, this paper contains little explanation of fuzzy logic and FLC. For more information, Sugeno (1985), Lee (1990a, 1990b), Mendel (1995) and Kantrowitz (1995) are good introductory sources. What follows is a very brief overview of fuzzy logic and FLC, given as a convenience to the reader, so that he or she does not have to refer to too many other outside sources to understand how the Fuzzy Logic Cutting Order (FLCO) program works.

Before explaining fuzzy logic, the concept of fuzzy sets must be introduced. Any discussion of sets assumes a "universe of discourse," which is simply a collection of objects. In ordinary set theory, an item from a universe of discourse, U, is either an element of a given set, or it is not. Therefore, the membership function of the set takes only two values, {0,1}, where 0 indicates nonmembership, and 1 indicates membership. A fuzzy set, by comparison, can be viewed as a generalization of this concept. A fuzzy set has a membership function that takes values in the interval [0,1], and any item *u* in U can be said to have some "degree of membership" in the set (Lee, 1990a).

A2.1 Fuzzy Set Defined

The following several definitions are taken primarily from Lee (1990a, 1990b), Dubois (1980) and Mendel (1995). A fuzzy set F in a universe of discourse U may be represented by a set of ordered pairs of an element, u, and its degree of membership function, $\mu_F(u)$:

$$F = \{ (u, \mu_F(u)) \mid u \in U \}.$$

U may be either discrete or continuous, and an example can be viewed graphically as in Figure 13 (here U is continuous).

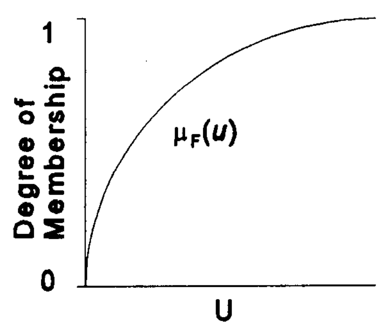

Figure 13 Example of a fuzzy membership function

As an example, consider a universe of discourse of "all people," and the set of "short" people. The membership function for an ordinary set, "short" might assign a value of 1 for all people strictly less than six feet in height, and 0 for all people greater than or equal to six feet in height. That is, all people less than six feet are members of the set "short," and everybody else is not. For most practical purposes however, little important difference exists between a person who is 5.99 feet in height and a person who is 6.01 feet in height. The problem is that "short" is not a sharply defined term--there are different degrees of shortness. Here, fuzzy sets prove useful because they provide a "basis for a systematic way for the manipulation of vague and imprecise concepts" (Lee, 1990a).

In comparison, consider a fuzzy set "short" whose membership function is defined as follows:

$$\mu_{short}(u) = \begin{cases} 1, & u \leq 4 \\ -\tfrac{1}{2}u + 3, & 4 < u < 6 \\ 0, & u \geq 6 \end{cases}$$

Here, a person who is 6.01 feet in height is not a member of the set "short," and a person who is 5.99 feet in height has a degree of membership of only -½(5.99) + 3, or 0.005. In other words, the person does not possess very much of the quality of shortness, but the fact that the person is shorter than the person who is 6.01 feet tall is reflected in their relative degrees of membership.

A2.2 Union and Intersection of Fuzzy Sets

If A and B are two fuzzy sets in U with membership functions $\mu_A(u)$ and $\mu_B(u)$ respectively (where $u \in$ U), then the membership function of the union of sets A and B is pointwise defined for all u as:

$$\mu_{A \cup B} = \max\{\ \mu_A(u),\ \mu_B(u)\ \}. \qquad (8)$$

Graphically, an example of this can be viewed as in Figure 14a, where the bold line represents the union of the two fuzzy sets A and B. The membership function of the intersection of sets A and B is pointwise defined for all u as:

$$\mu_{A \cap B} = \min\{\ \mu_A(u),\ \mu_B(u)\ \}. \qquad (9)$$

An example is shown graphically in Figure 14b, and the bold line represents the intersection of fuzzy sets A and B.

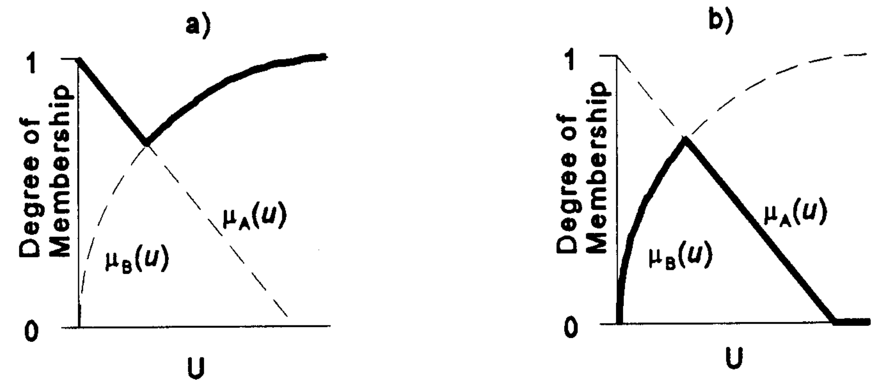

Figure 14a, b Examples of fuzzy union and fuzzy intersection

A2.3 Fuzzy Control Rules and Fuzzy Reasoning

As in traditional logic, a fuzzy logic conditional statement is of the form

IF (condition) THEN (consequence),

the difference being the condition and consequence rely on fuzzy sets. A fuzzy control rule (or simply "fuzzy rule") is a fuzzy conditional statement in which the premises are conditions in some controlled system, and the consequence is a control action in that system.

To see how fuzzy reasoning works in a control application, suppose an FLC system contains two fuzzy rules

R_1: if x is A_1 and y is B_1 then z is C_1;

R_2: if x is A_2 and y is B_2 then z is C_2.

Here x and y represent elements in the domains of some non-fuzzy inputs, z is an element in the domain of a non-fuzzy output and A_i, B_i and C_i (i = 1, 2) are fuzzy variables. The process of FLC takes place usually in four steps.

In the first step, fuzzification, the weights, or firing strengths of each rule are calculated by either intersection (Sugeno, 1985; Mendel, 1995)

$$w1 = \mu_{A1}(x^0) \wedge \mu_{B1}(y^0),$$
$$w2 = \mu_{A2}(x^0) \wedge \mu_{B2}(y^0),$$

or multiplication

$$w1 = \mu_{A1}(x^0) \times \mu_{B1}(y^0),$$
$$w2 = \mu_{A2}(x^0) \times \mu_{B2}(y^0).$$

Figure 15 is an illustration of fuzzy reasoning using the former. (Sugeno, 1985; Mendel, 1995) The top row of the figure represents R1 for some fuzzy sets A1, B1 and C1, while the bottom row represents R2 for fuzzy sets A2, B2 and C2. Since the rule premises are connected by an "and" they are combined by taking the minimum as shown in Equation (8), (9) on page 68.

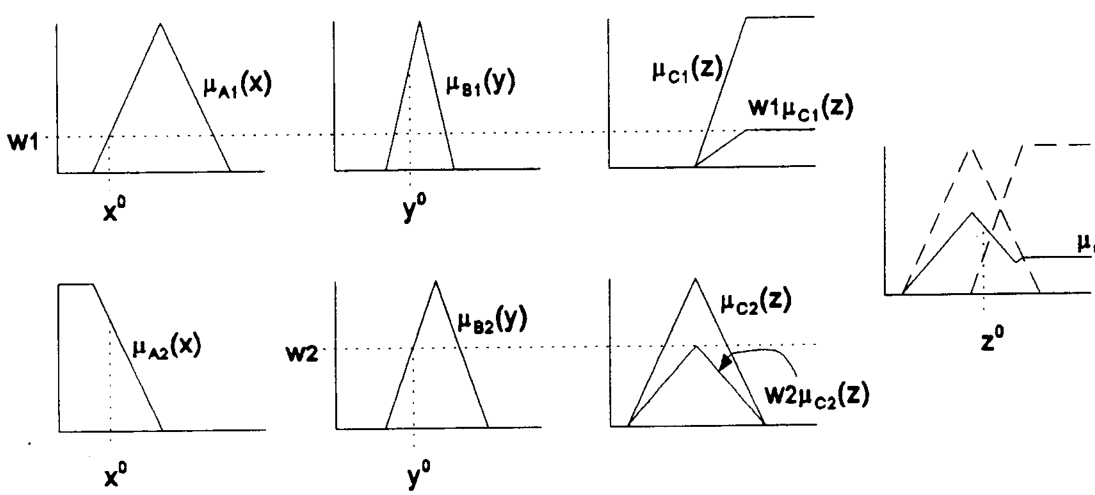

Figure 15 Example of fuzzy reasoning with two rules

In the second step, inferencing, the conclusions are weighted according to the firing strength of the premises, where:

$$(w1\mu_{C1})(z) = w1 \times \mu_{C1}(z),$$
$$(w2\mu_{C2})(z) = w2 \times \mu_{C2}(z).$$

This is an example of product inferencing. In minimum inferencing, μ_{Ci} would be truncated so that $\mu_{Ci}(z) \leq w_i, \forall z$.

In the third step, composition, these conclusions are combined to form a fuzzy set C* from which a control action is obtained:

$$C^* = w_1C_1 \cup w_2C_2.$$

While this example uses only two control rules, often many control rules are applied to the input, and in general $C^* = \bigcup w_iC_i, \forall i$.

In the fourth step, defuzzification, a crisp control action z^0 is found. Two methods of defuzzification include finding the point at which the value of $\mu_{C^*}(z)$ is maximized, and taking the center of area of $\mu_{C^*}(z)$:

$$z^0 = \frac{\int \mu_{C^*}(z)\ z dz}{\int \mu_{C^*}(z)\ dz}. \qquad (10)$$

The first method is sensitive to those rules that generate the largest degree of membership in the final fuzzy region, while the second method is sensitive to the height and breadth of the fuzzy region, and is the most widely used.