Real-Time Value Optimization of Edging and Trimming Operations for Rough, Green Hardwood Lumber

Daniel L. Schmoldt[a], Hang Song[b], Philip A. Araman[c]
[a]USDA Cooperative State Research, Education, and Extension Service, [b]Dept. of Computer Science, Arizona State University, [c]USDA Forest Service, Southern Research Station

Abstract

For edging/trimming operations in hardwood sawmills, an operator examines both sides—or just the waney-edged side—of each board, and makes a quick assessment of grade potential. The operator then decides where to edge and/or trim the board to achieve the intended grade and, presumably, maximum value. However, human operators can only achieve lumber values that are 62-78% of optimal. To achieve higher performance rates, computer-aided processing is needed to search through the millions of possible edging/trimming options. Using a computer to solve the edging/trimming problem means that board data must be provided from a scanning system. We have developed a prototype scanning system for rough, green hardwood lumber that can automatically describe a board and its defects. Those data are then used in a branch-and-bound (B&B) search for an optimal edging/trimming solution. In comparing this B&B algorithm with known maximal solutions for a suite of boards, we found that the B&B method attains the correct maximal solution for 92% of the boards and achieves 99% of the total maximal lumber value. Ninety-four percent of the boards attained the maximal value in less than 10 seconds (233 MHz processor). For all boards for which the algorithm obtained a maximal value, it did so in less than 8 seconds. Currently available computers can easily reduce that search time to 1-2 seconds per board. In comparing the B&B algorithm to the edging/trimming option in UGRS (the only other software that performs edging/trimming), our method has better accuracy (1-2% higher value recovery, 5-11% higher correctness) and is 40-100 times faster. The high value recovery of the B&B algorithm essentially eliminates that source of error for an automated edging/trimming system. Performance is therefore only limited by scanning accuracy.

1. Introduction

In hardwood sawmills, rough lumber is the principal product resulting from initial log breakdown. Lumber produced in sawmills is sold based on volume and grade, where grade is determined primarily by the percentage of defect-free cutting areas present on each board's surface. When logs are first cut into boards, they often contain residual bark areas (wane) and other undesirable characteristics (defects). Boards then go through edging and end trimming operations, where varying degrees of wane and some defects are removed. The resulting reduction in board surface area (volume) is often offset by an increase in grade from the removal of defect areas, producing a higher value board. Finally, edged and trimmed lumber is graded and sorted by grade or into grade groupings prior to kiln drying or sale.

In some sawmills, edger and trimmer operators visually examine the surfaces of each board for wane and defects and then make judgments about the placement of edge and trim saws to produce maximum valued boards based on their knowledge of lumber grades and on current lumber prices. Mill operators also instruct them on wane limits for boards. Optimizing the value of each board in this manner is a complex decision that is difficult even for trained operators with knowledge of hardwood lumber grading rules. In many cases, however, equipment operators rarely know current lumber prices and are rarely trained lumber graders. In other mills, lumber is profile scanned at 4- or 6-inch internals to reconstruct an outline image of the board and any wane. Software then optimizes edge saw placement for mill-instructed maximum wane allowance in grade using this coarse profile information. No defect information is collected and formal lumber grading rules are not applied during grade-wane optimization.

Prior work has clearly demonstrated the advantages of automating edging/trimming operations to maximize lumber value, rather than volume. Those operators with some lumber grading experience perform better than their counterparts, but still much below maximal. For example, lumber value recoveries of 78%, 65%, and 62% of maximum were achieved in three hardwood sawmills tested by Regalado (7). Higher recovery in the first mill resulted from operator training as a lumber grader. Experiments also confirm that the accuracy of human lumber graders is well below 100% (8). Huber (2) tested six willing hardwood roughmill employees and found that their grading accuracy was 68%. Polzleitner and Schwinghakl (6) carried out four independent trials on human graders and observed an average performance of 55%. While surface measure is easy to calculate reliably, lumber grade is not. The latter is a complex decision that includes visual packing of clear-wood cuttings onto a board's surface. Each grade also has restrictions on cutting sizes and numbers of cuttings allowed on the board.

To improve edging and trimming operations, attempts are being made to improve both (1) operator performance through training, e.g., the Edging and Trimming Training Program (3) and the Ultimate Grading and Remanufacturing System or UGRS (5), and (2) real-time mill processing through automation. For the latter, collaborative research between the USDA Forest Service and the Department of Electrical Engineering at Virginia Tech University, Blacksburg, VA has developed a rough lumber scanning system (composed of laser-ranging and gray scale imaging) and integrated it with application software to maximize edging and trimming decisions in hardwood sawmills. Scanning software automatically identifies defects on imaged boards and then passes that board "description" to software that finds a maximum-value solution for edge and trim line placements. Scanning performance needs are relatively modest for edging/trimming maximization, as Regalado et al. (9) demonstrated that 88% of maximal value can be obtained by locating and sizing wane, decay, and knots only. No matter what algorithm is used for maximization, however, it will need to perform board grading as part of that method. A computer software version of the National Hardwood Lumber Association (NHLA) grading rules exists (4) and can be readily incorporated into a maximization algorithm.

The work reported here describes the development and analysis of software for maximum-value edging and trimming, which is designed to integrate with the concurrently developed scanner system To achieve this goal, we created: (1) an exhaustive-search algorithm that methodically searches through all possible combinations of edge and trim lines—this gives us guaranteed maximal solutions, albeit very slowly—and (2) a limited-search algorithm that runs very quickly, but is not correct 100% of the time. We compared these two algorithms and also

compared our fast, limited-search algorithm to the edging/trimming module in UGRS, as it is the only other software that performs edging/trimming maximization.

2. Conduct of the study

For this project, our materials consisted of 2 separate data files and existing computer software. The latter included the NHLA-based hardwood lumber grading program (4) and UGRS (5). These are described in the following subsection. Our methods included development of both exhaustive-search and limit-search algorithms for maximal edging/trimming. Details of those algorithms are presented briefly in the subsequent subsection.

2.1 Materials

2.1.1 Board data

Two sets of board data were used in algorithm testing. One data set consists of sixty-five, waney-edged boards that were originally hand-mapped and coded by Regalado (7). Board data are recorded and stored using the standard ASCII file format that is compatible with the grading program. Its format is described in both of the preceding references. This set of boards uses rectangular defects at 1/4 in. resolution.

Although UGRS was developed as a lumber remanufacturing training tool, we felt that it would be instructive to compare its performance to our limited-search algorithm. There are two caveats to the comparison experiment, though. First, UGRS was not intended to edge and trim waney-edged boards, such as the ones in the original, 65-board data set. Second, because UGRS was designed as a training tool, it only operates on its own data set of boards or on boards created graphically and interactively by the user. It was not able to read and process the 65 boards in our first data set. Therefore, the comparison of UGRS with our limited-search algorithm used a subset of boards in the Red Oak Lumber Databank (1) that is included with UGRS. This databank contains boards that have been previously manufactured into rectangular boards, so some initial edging and/or trimming was already applied. A subset of boards from each of the #1C, #2C, and #3C data sets was used in our tests.

2.1.2 Grading program

The computer implementation of the NHLA rules for lumber grading considers nine types of "defects": stain checks, sound knots, unsound knots, wane, pith, splits, holes, and decay. Seven grades are defined in decreasing order of quality: FAS, F1F, Selects, #1 Common (#1C), #2 Common (#2C) and, #3 Common (#3C). Nevertheless, in this project, we used current industrial grade definitions for the central and southern Appalachian regions, where only 4 grades are considered: F1F (FAS on the best face and #1 Common or better on the second face), #1 Common, #2 Common, and #3 Common. Several changes were made to the grading program to accommodate these grades, primarily by editing it's configuration file.

It is possible to call the grading program as an executable stand-alone program with input and output exchanged through data files. However, it is more expedient to imbed the grading program's subroutines into another executable (i.e., our maximal edging/trimming programs) and

use the grading program's data structures to pass and return data, thus avoiding CPU-cycle intensive reads/writes. An entry-point subroutine can then be called, which in turn uses the remaining routines to evaluate the data and assigns a grade to the board, in question.

2.1.3 UGRS

The Ultimate Grading and Remanufacturing System (UGRS) is a computer program for grading and remanufacturing lumber. It is an interactive program that can both grade boards and remanufacture them for maximum value. UGRS consists of two modules: a grading module and a remanufacturing module. It uses the same grading program, described above, in both its grading and remanufacturing functions

For remanufacturing, UGRS first checks the board's original grade and its associated value. If the board grades as FAS (the best grade), then UGRS will not search further because the board's value cannot be increased. In all other cases, an attempt is made to obtain a remanufacturing solution that produces lumber with a higher total value and, when specified, a higher grade. Sometimes, UGRS generates two or more remanufactured solutions, which have the same value. The solution with the largest percentage of the original board's surface measure is retained.

UGRS includes four remanufacturing options. We decided to use the "Edging and Trimming" option as it most closely matches our edging/trimming approach, in that it produces a single board. In this case, UGRS attempts to rip and/or crosscut from the edges and/or the ends of the boards to increase the board's value. This remanufacturing option works well when defects are along the edges or ends of boards, especially for wane areas.

UGRS provides a high level of human interaction through override of grading decisions and manual remanufacturing. However, it was not designed to operate in real time as part of in-line mill processing. Still, based on results from a suite of oak boards, it has been shown that UGRS's solutions are the same or better than those obtained with earlier grading programs (5). UGRS also has a processing speed at least 50 times faster than earlier grading programs due to a better cutting unit algorithm.

2.2 Methods

All computer programming, program testing, and analysis were carried out on a 233 MHz PC running Windows 95[1]. Borland C++ 5.0 was used as the programming environment.

2.2.1 Exhaustive search algorithm

This search method tries all possible combinations of edging and trimming lines, within limits determined by the original grade and size of the board. It is guaranteed to find the maximal solution. Each setting of edging and trimming lines determines the surface measure (SM) of the board. Information regarding board defects is then passed to the grading algorithm (the embedded grading program), which provides a lumber grade for that board. The combination of grade and SM determines the board's value. Maximum value can be obtained by more than one combination of sawlines, although no preference was assigned to one maximal solution over another. Determining maximal edge/trim saw line placement in this manner can be very time

consuming computationally because each call to the grading algorithm may require substantial computation—for some boards, the edge/trim program may call the grading routine more than one million times and each board may require several hours of computer processing.

To estimate the maximum solution for a board, we performed six steps:

1. Obtain the grade and size of the board after applying the 50-50 wane rule. This initial solution is the current best value.

2. From the initial grade and size of the board, set the limits of edging and trimming.

3. Iteratively generate combinations of edging and trimming lines within the predetermined limits.

4. Evaluate the grade and volume of each edging and trimming line combination.

5. Replace the current best value with the current edging and trimming solution if it generates a better value.

6. Return the final maximum value and corresponding edging and trimming line solution.

Edging and trimming line coordinates were initially set at the lowest values in the range of cutting line variation (these ranges are discussed below). Edging lines were varied using quarter-inch increments. For trimming lines, one end uses quarter-inch increments and the other end uses whole-foot measurements because we assume that when boards are trimmed, their lengths still measure in whole feet.

To start edging and trimming maximization, we must first determine the outer edging limits by-using the 50-50 wane rule. Wane is not a true lumber defect; it only reduces the percentage of clear wood area on a board. In practice, mill operators tend to over-edge, that is, remove too much wood and wane during edging operations. Implementing the 50-50 wane rule allows us to mitigate that wasteful mill practice by considering edging solutions that leave more wood on the board. For implementation of the 50-50 wane rule, we stipulate that you cannot have more than 50% wane (in length) along either edge of the board. Furthermore, for the FAS-face, wane cannot extend more than 1/3 into the board width.

Even though this algorithm is intended to exhaustively enumerate all solutions to find the maximum value, several shortcuts were used to reduce the number of edge and trim line combinations. First, if the initial board graded as the highest grade (F1F), there's no need to search further by setting cutting line limits because a better grade and larger board are unattainable. So, the initial board value is the best. Second, while iterating through the edging/trimming line combinations, if a board is downgraded because of size, we can stop searching in that loop because the board can no longer achieve a better grade and its surface measure will always be less than the current board. Third, for different initial board grades, we can set different inner search limits. For example, if the initial grade of a board is #1C, we need to obtain a board whose grade is F1F to get a better value than the #1C board. Consequently, the inner limits for the edging and trimming lines are based on the F1F grade's minimum size: 6 in x 8 ft, instead of 3 in x 4 ft (minimum size for Common grades). If the current board's size is less

than 6 in x 8 ft, we cannot obtain a better grade than #1C, and so we do not need to search further. The lumber prices in **Table 1** were used in calculating lumber value.

Table 1. Prices for different grades ($$/BF) used in evaluating the algorithms

Grade	Price
F1F	1.12
#1C	0.78
#2C	0.46
#3C	0.385

The exhaustive search algorithm not only obtains the maximum value for each board, but it also finds (and records) the maximum value for each possible grade. Because the maximum value in each grade is the board with the largest SM in each grade, the program actually saves the largest SM for each. If we change the lumber price structure to conduct other analyses, we normally have to process each board again—which may take many hours of CPU time. By saving the largest SM for each grade, price changes only require checking the largest surface measure for each grade, and multiply it by the price to get the new maximum edging/trimming solution under that new price structure.

2.2.2 Branch-and-bound search algorithm

An exhaustive search procedure that searches for optimal edge and trim sawlines is not feasible for real-time applications because there are often hundreds of thousands of possible solutions to check. Each solution requires a board grading operation using the grading program. The total time spent in grading operations can amount to several hours. Our approach to this problem applies a more efficient search procedure based on the branch-and-bound (B&B) technique. It aims to greatly reduce the number of calls to the grading program, thereby greatly reducing the total search time.

By structuring the search space, it should be possible to ignore most inferior solutions and focus only on those candidate solutions that are likely to represent a maximal edging/trimming (E/T) value. In the B&B approach, the solution set is partitioned into several subsets, in a top down tree structure. Each of these subsets (subtrees) can be further partitioned as needed, or a subtree can be fathomed. A subtree is *fathomed*, when we can determine that further search on this subtree cannot achieve a better value than the current best value obtained. For our maximization problem, we have an initial *lower bound* for the maximum board value, which is the original value of the examined board (after the 50-50 wane rule is applied). This is our initial current best value, and any maximum E/T solution must be greater than or equal to this value.

As solutions are examined in each subtree, a solution that improves on the current best E/T value becomes the new lower bound. Additional branches are expanded until all existing branches are fathomed, then the current best solution is the best answer (maximal E/T solution).

There are two important observations that make the algorithm work efficiently.

1. The original board has the maximum SM and can be assigned an initial lumber value. As we move edging and trimming lines inward searching for better solutions, we monotonically decrease the SM of the board and concomitantly the value of the board. [Note: This monotonicity is a conceptual abstraction because SM is measured in integer units of board feet. On closer examination, it is really a decreasing step function.] This decreasing function is depicted in the graph of value vs. volume (SM) of **Figure 1.** At some point in this progression from right to left, we will encounter a grade change due to removal of some defect(s) or part of a defect(s) or due to a reduction in board size or percentage of clear wood cuttings (or some other grade-specific criteria). This grade change will result in a discontinuity in our graph and might result in a sudden increase in board value (if grade increases). In another cases, board grade may decrease, resulting in a sudden drop in board value. So, there is a monotonic decrease in lumber value as we move edge/trim lines inward until we reach a value discontinuity caused by grade increase or decrease. These discontinuities are the critical points that are very important in the B&B algorithm.

2. Board value decreases monotonically *after* application of the 50-50 wane rule and *after* each discontinuity. Therefore, the maximum board size for a particular grade must occur at a critical point. All other potential solutions on the graph must necessarily be less.

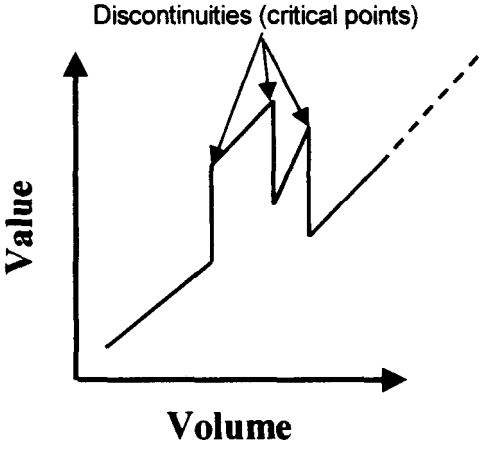

Figure 1. Volume vs. value plot depicts discontinuities in the solution search.

These two observations suggest that an efficient search algorithm can be developed, if (1) we can find the critical points efficiently and reliably and (2) only a small number of critical points must be examined.

Our implementation of the B&B algorithm begins with an initial solution that represents our best solution so far (lower bound). In the E/T problem the initial solution is to place the edge/trim lines near the edges and ends of the board based on the 50-50 wane rule. From that

initial solution we construct a tree of all other possible solutions. Each time a new node is expanded, four branches are created; these correspond to the inward movement of the four edge/trim lines from their current placements to the next critical points. In fact, our search for critical points will ignore those points that downgrade a board because improvements in value can only occur with grade increases.

In a hypothetical example, at Level 0 of the B&B tree **(Figure 2)** we grade the initial board (based on the 50-50 wane rule) and find that it grades as #2C. We label that starting node #2C and record that grade and value as our best solution (the lower bound on the maximum solution). Four nodes are generated at Level 1 by expanding the node in Level 0 to the next critical points from each current edge and trim line inward: left trim (LT), right trim (RT), top edge (TE), and bottom edge (BE). We quickly estimate whether the remaining board (for each of those four nodes) could possibly still grade higher than #2C, the current best grade. [Recall that we are interested only in critical points that increase grade.] We can use minimum board size to make this determination (or if the expanding node's grade is the highest grade, no higher grade is possible). If any of these nodes do not meet that criterion, then the branch headed by that node is bounded and we will label it as fathomed (marked with an "X" in the figure). The LT and TE nodes generated at Level 1 do not have critical points with an increase in grade, so those branches are fathomed. Because no solutions on this branch can exceed the current best grade of #2C, none can improve on the best solution —each has a SM that is strictly less than the current best solution (our lower bound on the maximal). If a node in Level 1 has a better solution than the current best value, e.g., the RT node, we set the current best value to this node's value and check if this node is fathomed. That is, because the RT node grades as #1C, we check to see if its board size will still allow for a higher grade (for this example, we find that it is large enough). However, the BE node grades as F1F and, in this example, it turns out that this board has a larger SM than the #1C node. So, we can now fathom the #1C node because even if its branch can achieve an F1F grade, the board will be smaller and have lower value. Because our F1F node is the largest possible board of that grade, that branch is fathomed and the current best solution (from that node) is our maximum solution.

In general, for each of the unfathomed nodes remaining in Level 1 we would branch again to Level 2 and examine these nodes. Each node in Level 2 represents two inward movements of edge/trim lines (one from Level 0 to Level 1 and one from Level 1 to Level 2). When all branches of the tree have been fathomed, the current best solution is the maximal value.

While this seems like a very straightforward approach to solving the edging/trimming problem the part that we have not yet defined is how we determine critical points. Critical points are located by moving edge/trim lines inward to remove part (or all) of a grading defect. This means that a critical point may be anywhere around a defect. It remains to locate it as precisely as possible. First, a defect is chosen such that its outward extent is the closest to the current edge/end of the board. Second, the critical point search range is this defect's length or width (depending on whether we are moving edge or trim lines). Third, we use a binary search to find the critical point. Initially, we set the cutting line at the mid-point of the defect. There are then 3 cases to consider:

1. If this initial placement produces a better value (and a grade increase) than the current best value, we move the cutting line halfway toward the outside edge of the defect to retain more surface measure. This produces a second placement of the cutting line.

2. If the initial placement produces a value that is less than the current best value and there has been no grade increase, we move halfway toward the inside edge of the defect to remove more defect (resulting in a second placement).

3. If the initial placement produces a value that is less than the current best value and there has been a grade increase, we locate our second placement halfway toward the outside edge of the defect (i.e., a grade increase has occurred, but our initial placement reduced the SM too much, so we want to move back outward to capture additional SM).

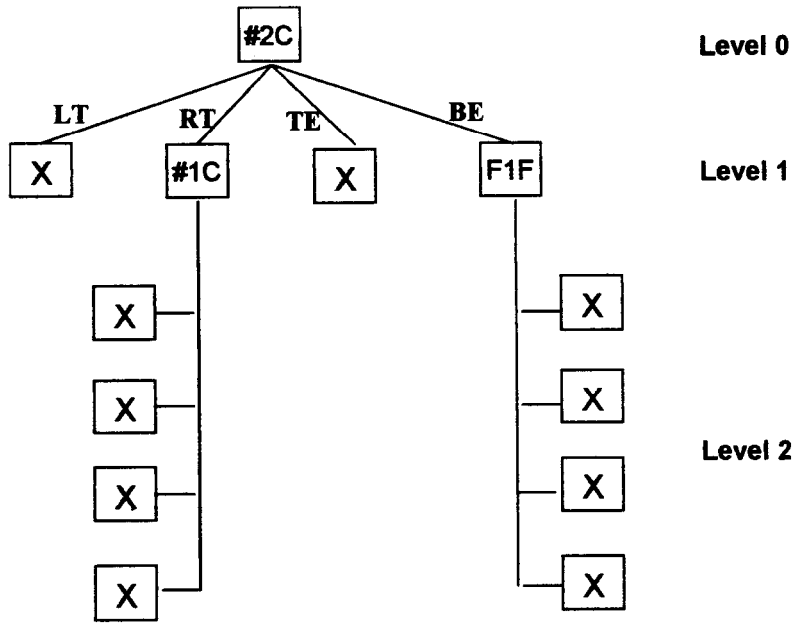

Figure 2. An example branch-and-bound search tree

As the binary search continues (replacing "initial placement" with second, third, fourth, etc.), we locate successive placements halfway between the last placement and the most recent inner or outer placement. Whether an inner or outer placement is chosen depends on the conditionals above. If, after searching through the entire defect, we get a better value than the current best value (and better grade), we establish a new node in this cutting direction and set its value to that obtained for this critical point. If a better value is not obtained, we continue searching the next defect in this direction for a critical point, or until the minimum size rule is violated for the current best value's grade. In the latter case, this branch is fathomed.

This procedure circumvents the use of the grading program whenever possible. It does this by keeping track of (1) the current best value board (its value, grade, and SM,) and (2) current board size. The B&B can then determine when a particular branch of the solution tree cannot possibly

improve on the current best value. Most of the algorithm's effort occurs in the critical point search, where most of the board grading calls are made.

3. Results and discussion

Using the procedures described in the preceding sections, a maximum edging and trimming solution (maximum value) can be found for each board and also a reduced-search solution (which may or may not be maximal). In this section, we compare: (1) the exhaustive search and B&B algorithms and discuss the advantages of the B&B and why it does not obtain the maximum value in some cases and (2) compare the B&B and UGRS algorithms with respect to speed and accuracy.

There are two ways to check the performance of the B&B algorithm. One, we compare the B&B-obtained value for each board with the maximum value (exhaustive search), to see what percentage of boards actually obtain the maximum. Two, we compare the total value of edged/trimmed boards for the B&B and the exhaustive search algorithms to measure total value recovery. From an industry standpoint, the latter measure is the most critical.

Accuracy is not the only factor considered, however. The other one is processing speed. The exhaustive search algorithm is the most accurate, but it is useless for real-time applications, where software processing speed must mesh with lumber processing speed. For industrial use, we need to find a maximal E/T solution for any board in just a few seconds. This means that we may need to sacrifice some loss of accuracy to obtain a solution quickly. Therefore we also need to analyze algorithm performance with respect to processing speed.

3.1 B&B versus exhaustive search

The 65 boards in our board data file were first processed using the exhaustive-search algorithm. The combined maximum value of these 65 boards is $395.27. The search required a total of 30,335,357 calls to the grading routine, for an average of more than 460,000 calls per board. Next, the 65 boards were processed using the B&B algorithm. The total value of these boards using the B&B search is $391.00. Therefore, the value recovery is 391.00/395.27 x 100% = 98.9%. The accuracy rate is 60/65 x 100% = 92.3% of the boards achieved the maximum value. The total number of calls to the grading routine by the B&B algorithm is 5264 and the total processing time is 253 s. The B&B algorithm reduced the number of the grading program calls by a factor of 6000 less than the exhaustive search.

As we look more closely at search performance, we note a few additional things. Most B&B searches only extend down only 1 or 2 levels and less than 4 nodes total. Only 2 boards have 3 levels and 1 board has 4 nodes. This searching efficiency supports the search strategy we employed and its use of critical points.

There are 4 boards whose processing time exceeded 10 s, although all four of those attained the true maximum value. After generating the initial results, we processed those four boards again but limited the processing time to a maximum of 10 s. In each case, the maximum value did not change for each board, but the number of grading calls decreased markedly. This

suggests that even with a relatively slow CPU, if the maximal value can be found by the B&B algorithm, it can be found relatively quickly.

In **Figure 3,** we graph the percentage of the 65 test boards that obtained a maximal solution within a particular time limit using the B&B algorithm. From this illustration, we note that 83% can be processed in 5 s and a full 92% can be processed in 10 s. Four out of the sixty-five boards required more than 20 s. For each of those boards, the B&B algorithm spent a lot of processing time (and grader calls) evaluating left and right trim positions.

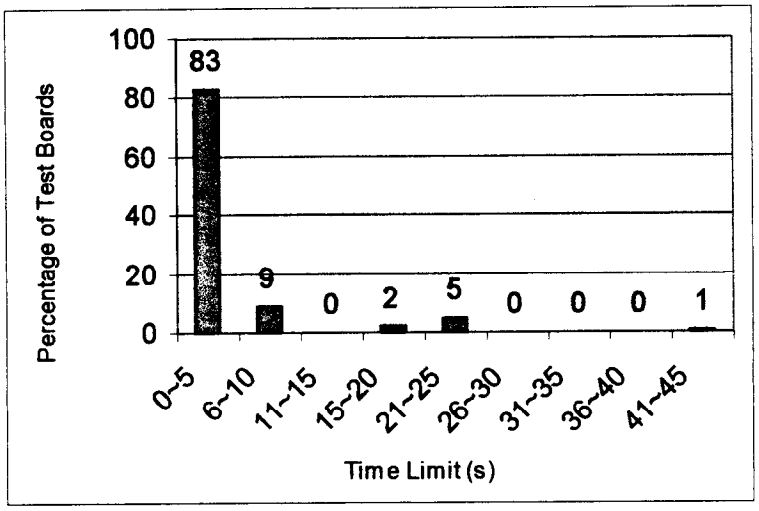

Figure 3. The number of waney-edged boards (expressed as a percentage) that completed processing in a given period of time.

3.2 B&B versus UGRS

Results of the comparison of the B&B algorithm and UGRS are summarized in Because UGRS does not include the 50-50 wane rule, it can reach incorrect results for some boards. For most boards in the Red Oak Databank, there is not much wane on either edge, so the 50-50 wane rule is not needed and so was not included in the software. But, for some boards (about 5% in the databank), significant wane is present and including the 50-50 wane rule can affect the results.

For the 87 boards whose original grade is #2C, the B&B algorithm is about 40 times faster than UGRS for the suite of boards. For the B&B algorithm, this averages out to about 0.5 s/board. Again correctness of the B&B algorithm is much higher than UGRS, while value recovery is only slightly better. Because the #2C and #3C boards have been previously manufactured, they lack significant end defects. The absence of end defects greatly increases processing speed. As we noted earlier, end defects are where the B&B algorithm spent most of its processing time for the waney-edged boards.

Table 2

The 177 boards whose original grade is #1C can only be improved, of course, by upgrading the board to F1F. Consequently, the B&B algorithm runs very fast—about 0.25 s/board. The B&B algorithm produces substantially higher correctness (number of boards with the correct maximum value) and slightly higher value recovery than UGRS. Of the 5 boards that the B&B algorithm missed, 2 were common to the 17 that UGRS missed. While the B&B algorithm and UGRS are very close in value recovery (1 cent on each dollar), the B&B algorithm has three times fewer errors and is almost 100 times faster.

Because UGRS does not include the 50-50 wane rule, it can reach incorrect results for some boards. For most boards in the Red Oak Databank, there is not much wane on either edge, so the 50-50 wane rule is not needed and so was not included in the software. But, for some boards (about 5% in the databank), significant wane is present and including the 50-50 wane rule can affect the results.

For the 87 boards whose original grade is #2C, the B&B algorithm is about 40 times faster than UGRS for the suite of boards. For the B&B algorithm, this averages out to about 0.5 s/board. Again correctness of the B&B algorithm is much higher than UGRS, while value recovery is only slightly better. Because the #2C and #3C boards have been previously manufactured, they lack significant end defects. The absence of end defects greatly increases processing speed. As we noted earlier, end defects are where the B&B algorithm spent most of its processing time for the waney-edged boards.

Table 2. Correctness (%), value recovery (%), and processing time (minutes) are shown for the B&B algorithm and UGRS with each board data set. The number of boards in each data set appears in parentheses.

	#1C Boards (177)			#2C Boards (87)			#3C Boards (95)		
	Correctness	Value Recovery	Time	Correctness	Value Recovery	Time	Correctness	Value Recovery	Time
B&B	97.2	99.4	<1	92.0	98.7	<1	92.7	99.0	2.1
UGRS	90.4	98.4	65.4	80.5	96.3	32.8	87.4	98.3	170

For the 95 boards whose original grade is #3C, the B&B algorithm is about 82 times faster than UGRS. Because the 50-50 wane rule has less influence on edge/trim lines placement for #3C board (containing many defects), the accuracies of the B&B algorithm and USRS are much closer than for the other data sets. The slim difference in value recovery remains about the same, though. Of the 7 boards that the B&B algorithm missed, 5 were common to the 11 that UGRS missed.

While UGRS ran much slower, it was also necessary to restrict UGRS to a time limit, so that all the boards could be processed in a reasonable amount of time. The time limit was set at 300 s. This means that UGRS stops if the time exceeds 300 s and the program returns the best value at that time. Consequently, the total processing time for UGRS is an underestimate of the time it would have used if no time limit was set. For the B&B algorithm's processing time, we noted

that about 96% of the boards were processed within four seconds. Only four out of the 95 boards required more than 10 s.

4. Conclusions

One reason that the B&B algorithm cannot find the maximum solution for some boards is that critical points cannot be positioned accurately for long, trimming defects. In this project, we use a binary search to find critical points. Binary search performs well when the defect is short but for long defects, a binary search can be inaccurate. The decision points used in our binary search (to decide in which direction to move next) are not infallible, and so occasionally result in some poor choices. Nevertheless, these "bad" choices do not result in a great value loss for any board. The maximum *grade* is still found, but the resulting board is submaximal.

While examining the exhaustive-search solutions for several boards for which the B&B algorithm did not attain the maximal value, we noted a flaw in our theoretical approach. That is, for end trimming, the true critical point that we seek is obtained by moving both trim lines simultaneously. Our approach moves one trim line at a time. Whereas, by moving both trim lines inward simultaneously a shorter distance, a better solution can be attained than by moving them independently and a greater distance. We can envision a binary search method that attempts to move both lines simultaneously, but we suspect that, given our current high value recovery, such a search would suffer from a greatly diminished return for greatly increase computational effort.

When board quality diminishes, as in #2C and #3C, there will always be a few boards that take a very long time to process. Most waney-edged boards (about 94%) can each be processed within just 8 s with a 233 MHz Intel CPU; most are processed in less than half that time. If the CPU speed is increased (current CPU speeds are up to 1.7 GHz), performance of this program will be proportionally faster. Still, it is possible to encounter boards that may take 40 s or more (current CPU timing) to process. Because we cannot guarantee that the B&B algorithm will find a maximal solution in a set amount of time, an operational version of this software will need to include a processing time cut-off value that provides the current best solution at an upper time limit. Given that for the current data set all correct solutions were obtained in less than 10 s, the use of a cut-off time value should not adversely affect accuracy.

The B&B algorithm is accurate, though not perfect. From the data reported here, we know that the B&B algorithm will recover more than 98% of the maximum edging/trimming value. In addition, the percentage of boards for which the B&B algorithm attains the correct maximum value is more than 92%. Comparatively, current, manual operations are only able to achieve approx. 65% of maximal value. The analyses conducted here assume that the board data are accurate, whereas any scanning system that provides real-time board data will not be 100% accurate. But, because the optimization software is >98% accurate, operational edging/trimming accuracy will then be determined primarily by scanning accuracy. The software reported here, then, has largely eliminated one source of error for optimal edging/trimming automation.

5. References

1. Gatchell, C. J., J. K. Wiedenbeck, and E. S. Walker. 1993. A red oak data bank for computer simulations of secondary processing. Forest Products Journal 43(6): 38-42

2. Huber, H. A., S. Ruddell, and C. W. McMillin. 1990. Industry standards for recognition of marginal wood defects. Forest Products Journal 40(3): 30-34

3. Kline, D. E., E. M. Wengert, P. A. Araman, and P. Klinkhachorn. 1992. Hardwood lumber edger and trimmer training system. Forest Products Journal 42(1): 53-57

4. Klinkhachorn P., J. P. Franklin, C. W. McMillin, R. W. Conners, and H. A. Huber. 1988. Automated computer grading of hardwood lumber. Forest Products Journal 38(3): 67-69

5. Moody, J. C., C. J. Gatchell, E. S. Walker, and P. Klinkhachorn. 1998. An introduction to UGRS: the ultimate grading and remanufacturing system. Forest Products Journal 48(9): 45-50

6. Pölzleitner, W., and G. Schwingshakl. 1992. Real-time surface grading of profiled wooden boards. Industrial Metrology 2(3/4): 283-298

7. Regalado, C. 1991. Optimization of edging and trimming operations for red oak lumber. M. S. Thesis, Virginia Polytechnic Institute and State University, Blacksburg VA

8. Regalado, C., D. E. Kline, and P. A. Araman. 1992. Optimum edging and trimming of hardwood lumber. Forest Products Journal 42(2): 8-14

9. Regalado, C., D. E. Kline, and P. A. Araman. 1992. Value of defect information in automated hardwood edger and trimmer systems. Forest Products Journal 42(3): 29-34

[1] Tradenames are used for informational purposes only. No endorsement by the US Dept of Agriculture is implied.

SCANTECH 2001

THE NINTH INTERNATIONAL CONFERENCE ON SCANNING TECHNOLOGY AND PROCESS OPTIMIZATION FOR THE WOOD INDUSTRY

PROCEEDINGS

NOVEMBER 4-6, 2001

HOLIDAY INN
SEATTLE-TACOMA INTERNATIONAL AIRPORT
SEATTLE, WASHINGTON, USA

SPONSORED BY
WOOD MACHINING INSTITUTE

IN COOPERATION WITH

FOREST PRODUCTS SOCIETY
AND
INTERNATIONAL UNION OF FOREST
RESEARCH ORGANIZATIONS