Adaptive Optimization Methods to Improve the Performance of Chop Saw Systems

by

Lin Ye

B.Eng., Nanjing Forestry University, China, 1982

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
MASTERS OF SCIENCE

in

THE FACULTY OF GRADUATE STUDIES
(Department of Wood Science)

We accept this thesis as conforming
To the required standard

THE UNIVERSITY OF BRITISH COLUMBIA

April 2004

Library Authorization

LIN YE
Name of Author *(please print)*

21/04/2204
Date (dd/mm/yyyy)

Title of Thesis: Adaptive Optimization Methods to Improve the Performance of Chop Saw Systems

Degree: Master of Science in Forestry Year: 2004

Department of Wood Science, Faculty of Forestry
The University of British Columbia
Vancouver, BC Canada

ABSTRACT

Optimizing chop saw is one of the primary methods for adding value to rough lumber (boards) and it is common to many breakout lines for plants processing solid wood. Because of the variety of boards to be chopped in real time and the diversity of parts (products) required on a cutting list, there are numerous ways of cutting the boards.

The purpose of this thesis was to advance the optimization algorithms that can be used in real-time to generate optimal cutting patterns, aiming to minimize the overall cost of the chop saw operation. Adaptive methods were used in the models to acquire the knowledge on the quality of boards and the inventory level for each part in real-time.

The first part of the project developed a real time goal-seeking model for chop saw optimization systems that can be implemented as an on-line control system. Two prioritizing strategies were used in the model by applying an adaptive method that can link the part value with the production level for each part during the production process. A simulation study was conducted to compare the performance of the goal-seeking model with a published model. The result shows that by selecting a proper prioritization strategy, the goal-seeking model can achieve a lower level of overproduction, higher part yield and lower overall cost. The study also found that the goal-seeking model balances the production rates for each part on the cutting list over the production run.

The second part of the project developed a combined linear programming and dynamic programming model that used adaptive optimization methods. This model considers the part size and quantity in the cutting list and the grade of boards being processed and the production level for each part chopped in real-time. The simulation study shows that, different from goal-seeking model that requires selecting a proper prioritization strategy for a given cutting list, the combined model has a steadier control to overproduction and good performance against different cutting lists. However, the result is sensitive to the overproduction penalty assigned to the model.

Key Words: chop saw, adaptive, algorithm, dynamic programming, linear programming, optimization, prioritization strategy, simulation.

TABLE OF CONTENTS

Adaptive Optimization Methods to Improve the Performance of Chop Saw Systems

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENTS

There are several individuals in Faculty of Forestry who should receive full recognition and gratefulness for their contributions to this thesis project.

I am deeply grateful to Dr. Thomas Maness for being my project supervisor and providing guidance, support and funding for the project.

To other members of my committee - Dr. John Nelson and Dr. Robert Kozak – Thank you! Your assistance and guidance were critical.

Mr. Andre Schuetz deserves special acknowledgement for his assistance in the development and debugging of the simulation program.

CHAPTER 1

1.0 INTRODUCTION

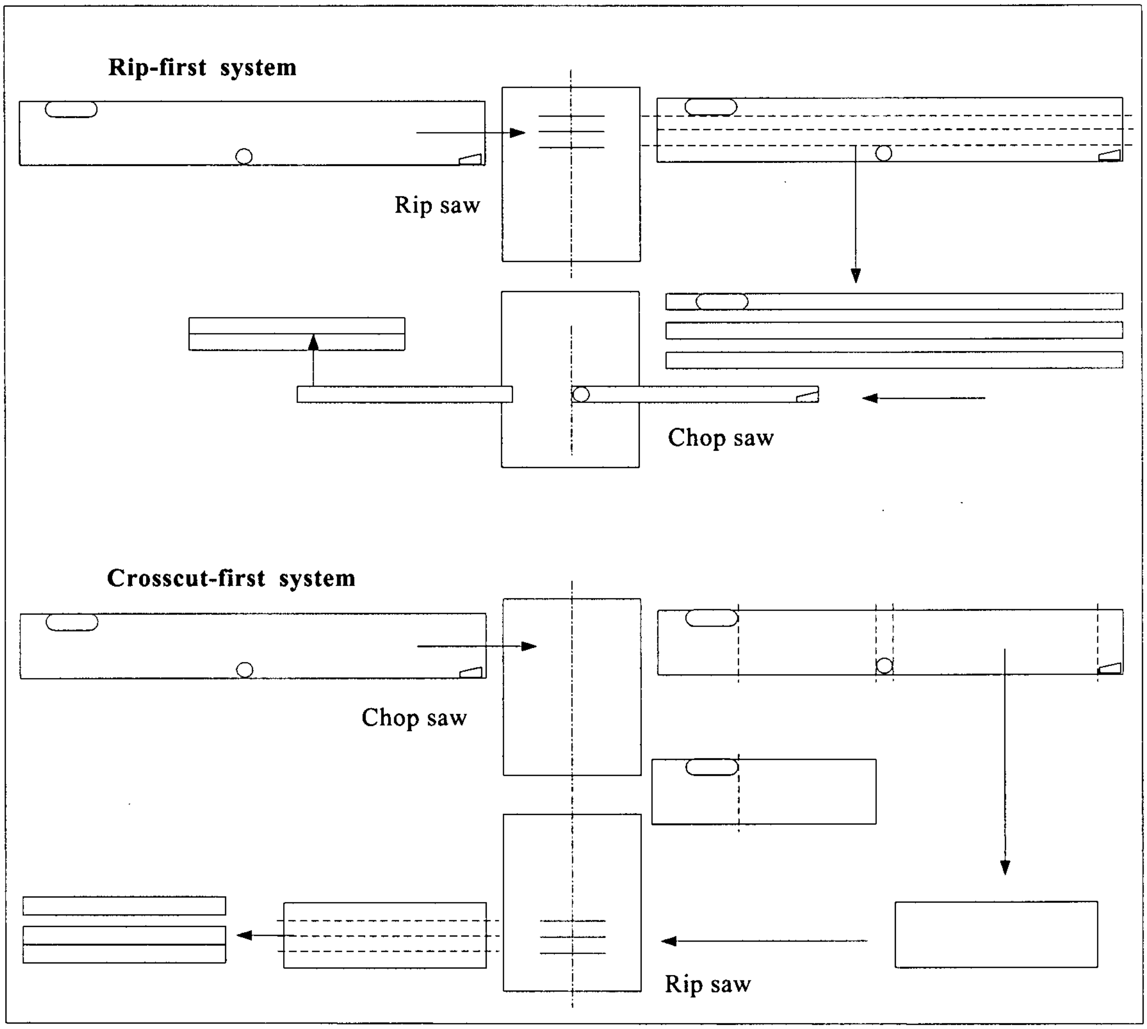
1.1 BACKGROUND

Cutting rough lumber into high quality short pieces of specific dimension parts is the first stage of adding value to lumber in secondary wood manufacturing. The parts are cut in the rough mill of furniture, cabinet or dimension parts plants. A list of required pieces, called a cutting list, describes the parts to be produced for a given production run. The manufacturing specification is contained in the cutting list, such as the length, width and quantity of parts required. The objective of a rough mill is to produce these parts on time and, ultimately, at the lowest overall cost (Buehlmann 1998).

The production processes used in a rough mill to produce parts are a reflection of the heterogeneous nature of lumber and therefore two basic types of sawing machines with allied equipment are employed: a rip saw and a chop saw. The rip saw consists of a horizontally aligned arbor equipped with a row of saw blades spaced appropriately in relation to the widths demanded and cuts the incoming lumber into boards along the direction parallel to its length. The chop saw consists of a saw moving upward or downward and cross cuts boards into the lengths as required on a cutting list.

Two different types of layouts for the sawing machines are used and they are defined as a crosscut-first system and a rip-first system. The distinction between these two systems is the sequence in which the rough lumber is cut to desired parts. Figure 1 shows the layout of both systems. The rip-first system cuts the incoming rough lumber into to long, narrow boards by the rip saw first, and then crosscuts the boards to parts in the chop saw. The crosscut-first system cuts the rough lumber to length first and thereafter to width. The selection of the systems basically is a function of the overall product strategy at the mill. A rip-first system is generally better suited to plants that use lower grade boards as raw material and basically produces narrow widths and long length parts. Plants that produce parts with wide widths and shorter average lengths are better served by crosscut-first system.

Figure 1. Rough mill configurations: rip-first vs. cross-first

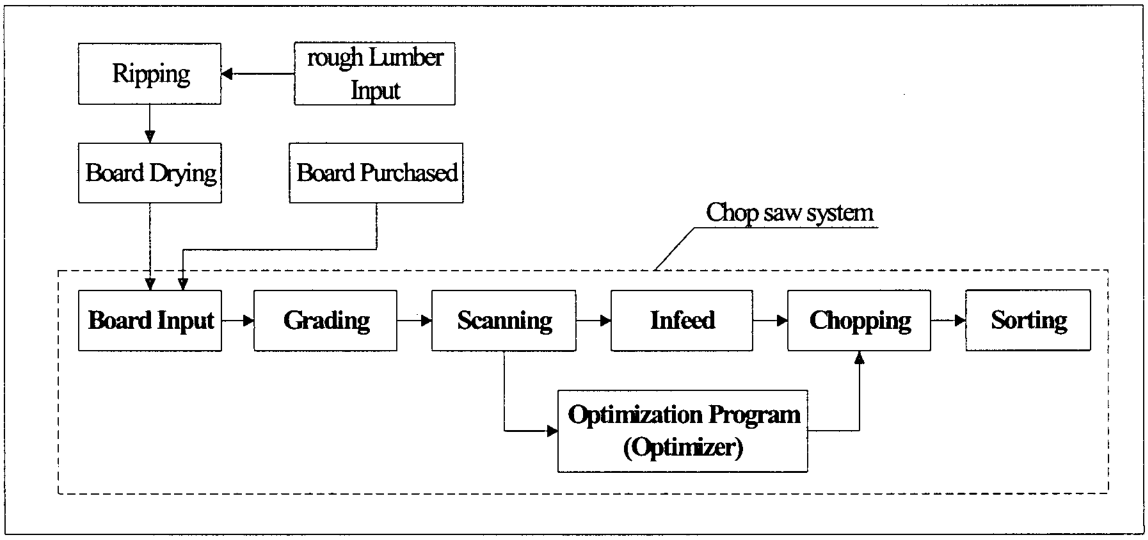


Although both systems are used in rough mills, rip-first systems are drawing more attention in the industry as 90% of all newly installed rough mill systems are predicted to be rip-first (Mullin 1990). The reasons for this development include 1) rip-first systems produce higher yields of longer parts from lower grades lumber (Gatchell 1987), 2) the pressure to utilize low grade lumber is increasing as higher grades are becoming less available, 3) the rip-first process can help operators to easily identify and locate the defects in boards, and 4) the process requires fewer and simpler operation decisions (Mullin 1990).

As a result of this increasing trend towards rip-first dominance, research to improve the rough mill operation has been focused on the rip-first rough mills. This study concentrates only on the chop saw process (crosscut process) of a rip-first operation.

The production sequence of a typical rip-first chop saw system for softwood lumber in British Columbia is shown in Figure 2.

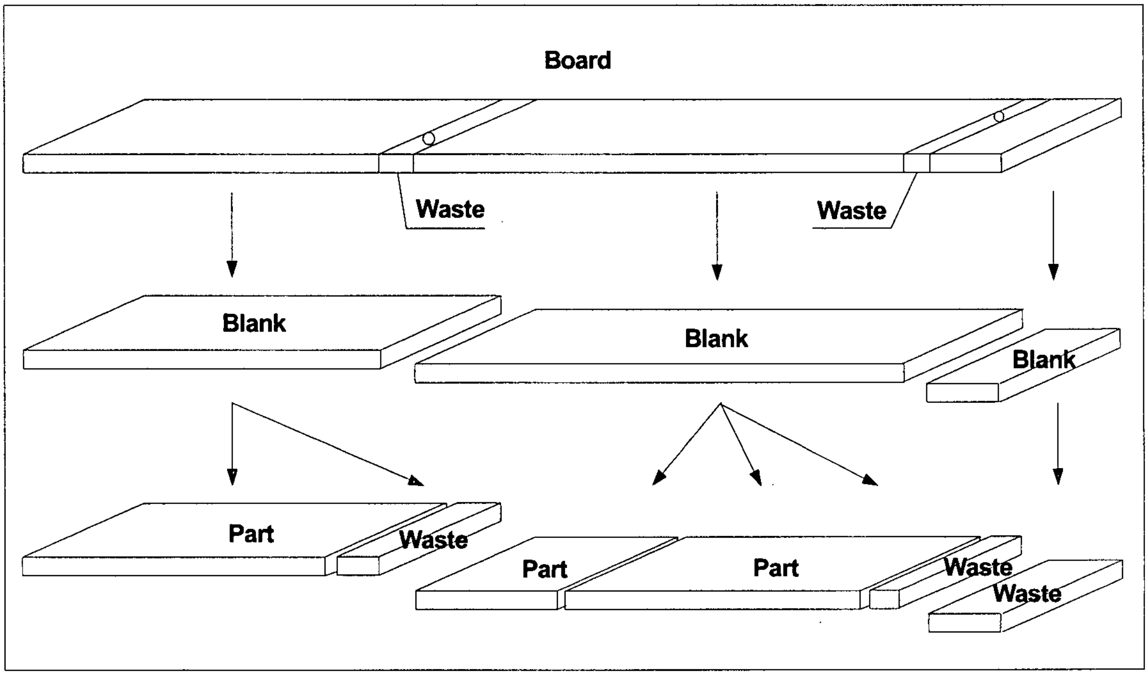
Figure 2. Process flow of a typical softwood chop saw system in British Columbia



The nomenclature of this study follows Figure 3 and is as follows. A *board* is a raw full length piece of lumber that has been ripped into the correct width in a previous step. A *blank* is the usable piece of lumber that exists between two defects. *Parts* are the pieces of lumber that are chopped from the usable blank.

Boards coming from either the rip saw in a plant or directly brought from outside sources are sent to the front of a chop saw system in the form of lifts which contain kiln dried boards with the same width and thickness, but different lengths. These boards are sent, piece by piece, to a defect marking station where a worker marks the positions of the defects on an incoming board with a fluorescent crayon or chalk by drawing lines perpendicular to the edge of the board. These lines, as shown in Figure 3, identify the waste, such as knots, wane, splits and other wood defects on the board.

Figure 3. Marking result on a board to identify the blanks from waste



The board is then fed through a measuring station equipped with encoders and fluorescent-reading cameras that measure the length of each blank and the location of the marks. This information is then passed to a computer where an optimization system, called “optimizer”, is contained. The function of the optimizer is to "remove" the defects and to calculate the optimal chopping patterns for the blanks (usable sections) on the board based on the cutting list and then to pass the decisions to the chopping mechanism control system before the board is chopped. This differs from a manual operation where the operator examines part of the board, makes a decision, positions the board over the saw, and chops.

The chopping mechanism consists of saw that can move upward and downward as the board is positioned longitudinally to chop out parts according to the orders from optimizer. The part chopped is subsequently pushed into a bin by one of a series of kickers installed at the out-feed area of the chop saw system. Once a bin is full, the parts are sent to inventory.

The task of a chop saw system for a rough dimension plant is to obtain all the parts on the cutting list at the minimum cost.

The optimization program in a chop saw system plays a critical role in this task. Due to the extensive input and output of the material through the system (Wengert and Lamb 1994), there exists great potential for the improvement of the program.

1.2 PROBLEM STATEMENT

Optimizing chop saws is one of the primary methods for adding value to boards. Because of the variety of boards to be cut and the diversity of parts to be produced, there are numerous ways to chop the boards. How well parts required by a cutting list can fit into boards with a specific quality is critical to the efficiency of a chop saw system. Many rough mills in the North America secondary wood manufacturing industry use chop saw optimization systems to optimize the cutting patterns.

The optimization process in a chop saw system consists of two nested optimization functions. The first is a blank optimization function in which either the volume or the value recovery from each blank is maximized. The second is a plant optimization function in which the plant must fill current orders from the factory on time at the lowest overall cost.

The overall cost associated with the chop saw operation is mainly composed of the following factors:

1. the raw material (lumber) cost,
2. the cost of holding inventory for the amount of parts overproduced,
3. the cost of underproduction (unable to fill up a cutting list). Generally this would be made up by purchasing a higher grade,
4. the cost of disposing the waste that cannot be sold, and
5. the processing cost.

Current chop saw optimization models used in industry operate on a single blank at a time. The optimization models are formulated as a "Knapsack Problem" which can be defined as (Definition 1):

> *"Find the optimal way to chop a piece of incoming blank on a board into parts described on a cutting list so as to maximize part yield or part value, or to minimize waste."*

As these optimization models (Gilmore and Gomory 1966; Giese and Danielson 1983; Brunner et. al. 1990; Sessions et. al. 1988; Chow 1998) look at the problem on a blank level, they are blank-by-blank optimization models.

However, it has long been recognized that strict blank-by-blank optimization produces a sub-optimal solution for a production run since it does not consider the plant optimization problem. Typically a rough mill produces solid wood parts to fill current orders. This means that a fixed quantity of each part must be produced for the upcoming production run. Over-produced parts must be held in the inventory, which increases working capital, uses valuable floor space in the plant, and incurs handling and damage costs. Under-production can delay assembly or delivery of the final product. This delivery delay can in turn cause the loss of customer good will, and cause a delay in revenue that affects the plant's cash flow. Overall, excess overproduction and underproduction reduce profit to the plant.

The first generation of these models optimized each blank based on a static set of part values on the cutting list without considering the number of parts required and current production levels. To effectively manage these systems, the production supervisor constantly monitors the production and inventory levels and modifies the priority setting for the part values on the cutting list as required. In practice, this usually takes the form of changing the value of a part to zero once the required production level is met so as to switch the production to other parts.

Second generation chop saw systems have automated this supervisor checking procedure, and some have automated the shut-off procedure as described above. Others use proprietary algorithms (Chow 1998) that are based on forcing the cutting of priority parts until enough of them have been produced.

However, the common problem with these systems is excessive over-production for short parts once the cut order for all parts is met. This is because the short parts are always easier to produce than long parts. This not only increases inventory cost but also means that more raw materials are needed to fill the current order. Moreover, since some parts are finished first, the remaining parts on the cutting list become fewer and fewer as production progresses. The optimization program therefore has fewer parts from which to choose, which causes the part yield to drop. To prevent this unexpected situation, some plants have to stop the production once the quantities for short parts reaches the desired level, and then either buy the under-produced parts from alternative sources (at extra cost) or switch to a higher grade lumber (at extra cost) to produce the remaining long parts. These activities inevitably increase the raw material cost to fill a given order.

To solve the plant optimization function, some models considered the chop saw optimization as a typical "One Dimensional Cutting Stock" problem which can be defined as (Definition 2):

> *"Find the optimal way to chop incoming boards with* ***known quality*** *(the length distribution of blanks on the boards is known) into parts described on a cutting list so as to complete the order requirement while maintaining lowest raw material cost or highest part yield."*

Models with this approach (Gilmore and Gomory 1965; Mendoza and Bare 1986; Foronda and Carino 1991; Carnieri et. al. 1992; Ghodsi et. al. 2003) provide a theoretical overall chopping solution to the boards by allocating different parts to deferent blanks. These models assume that the quality of the board is known. In other words, the distribution of blanks derived from boards is known before the production run starts. Obviously, they cannot be applied to a real-time chop saw process since the full knowledge of the distribution of blanks is only available at the end of a production run. In addition, the overall cost was not addressed in these models.

There are several other models, such as that by Thomas (1996), which define the chop saw optimization problem as (Definition 3):

> *"Find the optimal way to chop incoming boards with **unknown quality** (the lengths distribution of blanks on boards is known) into parts described on a cutting list so as to complete the order requirement while maintaining lowest raw material cost or highest part yield."*

Unfortunately, the overall cost was not considered in this model. In addition, the model uses an exhaustive search method to find the optimal solution, which limits the model's application in a real-time process.

The chop saw optimization problem for this thesis is defined as (Definition 4):

> *"Find the optimal way to chop incoming boards with **unknown quality** (the lengths distribution of blanks on boards is unknown) into parts described on a cutting list so as to complete the order requirement at the lowest **overall cost.**"*

Currently, there are no existing models to solve this problem. New models need to be developed that can solve both the blank optimization function and the plant optimization function simultaneously in real-time. And above all, overall cost should be used as a benchmark to evaluate the performance of new models.

1.3 OBJECTIVES OF THE THESIS

The purpose of this study was to advance the optimization algorithms for chop saw systems by developing two optimization models that solve the blank optimization function and the plant optimization function simultaneously using the least overall cost approach. Adaptive methods and dynamic programming were used so that the models could be implemented in real-time. The specific objectives of this research were:

1. develop a real-time optimization model, called a goal-seeking model, that can use different dynamic prioritization strategies to adapt to the knowledge of inventory levels for parts produced at real-time, aiming to optimize the utilization of raw materials and reduce the amount of overproduction for a given production run,

2. compare the goal-seeking model with Thomas's model (1996). The overall cost is used as a benchmark to evaluate performance,

3. develop an overall cost minimization model, called the combined model, that can use adaptive methods to get the knowledge on the grade of incoming boards and the inventory levels for parts produced in real-time, and

4. compare the performance of the combined model with the goal-seeking model using the overall cost.

1.4 ASSUMPTIONS USED IN THE THESIS

The research was based on the following assumptions:

1. all the data used in this study (such as costs, penalties, blank length distribution, cutting lists) reflect the real situation in a rough mill or a dimension part plant.
2. the relationships between the variables in the combined model are linear.

1.5 ORGANIZATION OF THE THESIS

A literature search on the previous developments in the field is found in Chapter 2. The goal-seeking model and the combined model developed in this study are described in Chapter 3 and Chapter 4, respectively. The methods are addressed in detail in these two chapters. In Chapter 3, a comparison of the goal-seeking model developed in this study with Thomas's model (1996) is conducted, along with the results and the conclusion. In Chapter 4, a comparison of the combined model developed in this study with the goal-seeking model is presented. The research is summarized and the recommendations for the future research are contained in the last chapter.

CHAPTER 2

2.0 LITERATURE REVIEW

This chapter first provides an overview of the existing optimization models that maximize either part value or part yield for chop saw systems. Thereafter, models to minimize either raw material cost or the amount of waste are described.

2.1 MAXIMIZATION MODELS

These kinds of chop saw optimization systems are basically divided into two categories: value maximization and yield maximization.

2.1.1 Value Maximization

One approach to the chop saw optimization problem is the value maximization model. The model puts a dollar value on each part in the cutting list. The problem is solved by analyzing each blank on an incoming board to produce the part lengths that create the most value.

The typical model of this kind was developed by Chow (1998). In this model, twenty different prioritization strategies were simulated for achieving maximum parts value recovery for a cutting list. The prioritization strategies used are based on the dollar values assigned to the parts to be produced. More desirable parts, especially long parts which are often difficult to get, are given considerably higher values than less desired parts.

However, the values assigned to the parts are often arbitrary as it is difficult to determine the actual value of a part flowing through a mill where the parts produced by a chop saw system are the raw material for the subsequent production processes. Therefore, the valuation strategy used in this model is dependent on the assumptions of the person running the system. In addition, although the part value can be maximized, the actual cost to achieve this value is unknown.

2.1.2 Yield Maximization

Since the raw material cost is by far the most significant cost incurred in a rough mill (Wengert and Lamb 1994), part yield is known to be the crucial factor in determining the profitability of a plant producing dimension parts. Using less raw material to meet a given order becomes one of the greatest challenges to the chop saw operation. Therefore, efforts have been directed at the development of yield maximization models.

Earlier models (Thomas 1962; Gilmore and Gomory 1966; Wodzinski and Hahm 1966; Stern 1978; Giese and McDonald 1982; Giese and Danielson 1983; Sessions et al. 1988) are based on simple placement algorithms that operate on a single blank at a time. These models defined the problem according to "Definition 1" described earlier. By calculating the area of each part, the model can determine the largest combination of part areas to fit for a given blank so that the largest part yield can be achieved from the blank. As these models lack the dynamic prioritization capabilities which consider the quantities of parts required and the levels of parts produced, they always result in over producing some parts and under producing others. Therefore, the solutions these models supply can only be considered as the sub-optimal. As these models only look at the optimal part yield at a blank level, they are often called blank-by-blank optimization models.

To overcome plant level sub-optimality, Dmitrovic (1992) and Thomas (1996) employed dynamic prioritization strategies in their models. These models are structured in two levels:

1. a placement algorithm to optimize the cutting patterns by looking at the chop saw problem at a blank level, and
2. a master algorithm to meet the cutting list requirement by looking at chop saw problem at a plant level.

The strategies were based not only on part size but also on current part quantity needed by a cutting list. With these dynamic strategies, part priorities are continually updated based on how many parts of each size remain to be cut. The goal of the model is to maximize the part yield while at the same time maintaining inventory at a low level.

Unlike the blank-by-blank optimization models, Dmitrovic's and Thomas's models defined the problem as "Definition 3", and therefore improved the part yield optimization by looking at the solution at both the blank level and the plant level. However, the benchmark for the models was the yield performance. How well the models performed in terms of the overall cost is unknown. Nevertheless, as an analytical tool, the models allow operators and researchers to examine the interactions and impacts of various rough mill operations.

2.2 MINIMIZATION MODELS

Models to minimize raw material cost or waste were another approach to solve the chop saw optimization problem. These models (Gilmore and Gomory 1965; Mendoza and Bare 1986; Foronda and Carino 1991) were based on a two-level hierarchical structure:

1. a sub-model considered as the Knapsack Problem to generate optimal cutting patterns for a given blank, and
2. a master model to look at the cutting list and optimize the overall production.

The sub-model and the master model interact iteratively until the problem is solved.

Carnieri and Mendoza (1994) developed a two stage model similar to above models for optimal cutting of lumber and composite panels into dimension or furniture parts. The objective of the model was to minimize the raw material cost.

In 2003, Ghodsi developed a rule-base expert system with an objective to minimize the waste for filling a given cutting list. In this model, either a crisp or fuzzy logic method is applied to prioritize or rank a particular cut sequence. The principle of the model is that it "learns" from the data in the cutting list by inspecting the lengths and desired quantities of each part and then ranks the parts based on their influence in inducing waste. The parts with larger lengths and greater ordered quantities are given higher priority. This iteration repeats until all parts are produced.

These models look at the chop saw problem as defined in "Definition 2". The assumption used in these models is that the raw material quality is known. In case of the composite panel cutting, these assumptions conform to the real situation as the quality and dimension of panels to be cut, for instance, 4 feet × 8 feet, are known beforehand. Therefore, the optimization problem associated with composite panels can easily be solved using this approach. However, the assumptions cannot be applied to chop saw production because the raw material quality is unknown. Since all the models mentioned above lack the dynamic capability to adapt to the knowledge of the raw material quality (which can only be known in real-time), they can not be used in a dynamic production process. Nevertheless, these models presented a sound theoretical approach to the chop saw optimization problem.

CHAPTER 3

3.0 GOAL-SEEKING MODEL

3.1 INTRODUCTION

As described in Chapter 2, Thomas (1996) employed dynamic prioritization strategies in his model. The model can automatically determine the part priority based on the number of parts required. Longer parts are weighted more heavily than shorter parts. The two dynamic part prioritization strategies simulated in this model were Simple Dynamic Exponent (SDE) and Complex Dynamic Exponent (CDE).

SDE prioritizes part length and width equally. The prioritization for SDE is:

$$SDE\ \Pr iority = (Length \times Width)^{WF} \quad [1]$$

$$\text{Where: } WF = (\sqrt{\ln(Need + 0.01)} \times MF) + 1.0 \quad [2]$$

CDE prioritizes part length and width separately. The prioritization for CDE is:

$$CDE\ \Pr iority = Length^{WF_{Length}} \times Width^{WF_{Width}} \quad [3]$$

$$\text{Where: } WF = (\sqrt{\ln(Need \times \max(1,(35 - Count)))} \times MF) + 1.0 \quad [4]$$

In [1] and [3], *Length* and *Width* represent part length and width respectively while *Need* in [2] and [4] is the current required quantity of a part.

WF in [1] and [3] represents a weighting factor obtained by applying the square root and natural log transformations to the needed part quantity. *MF* in [2] and [4] is a multiplication factor. The SDE strategy uses a *MF* value of 0.14 while CDE uses two *MF* values, 0.14 for length and 0.07 for width.

The difference between SDE and CDE is that with CDE strategy an extra priority is given to part sizes that have low initial requirements. *Count* represents the number of parts of a particular size that have been generated. In equation [4], this term places a preference on producing the first 35 parts of each size.

It was found that compared with static part prioritization strategies, the dynamic part prioritization strategies produced a cutting list using fewer raw materials. However, how well the model performs in term of overall cost is not reported.

In this chapter the goal-seeking model is described that addresses the problem defined in "Definition 4". A study was conducted to compare the goal-seeking model with the model developed by Thomas (1996). Overall cost is used as an indicator to evaluate the performance of the models.

3.2 MATERIALS AND METHODS

3.2.1 Raw Materials

Boards ready to be cut in a chop saw system come in the form of lifts (or packages) which contains boards with the same width and thickness, but different lengths. These boards are sent, piece by piece, to a defect marking station where a worker marks the defects on an incoming board with a reflective crayon by drawing lines perpendicular to the edge. These lines identify the waste from the blanks as shown in Figure 3. The scanning unit in the chop saw system detects the location of the marks on the incoming board and passes this information directly to the board optimization program. The task of the optimization program is to "remove" the defects and optimize the chopping of usable sections to get the highest combination parts based on the cutting list.

To simplify the simulation program, the raw material in this study is made up of blanks 100mm wide and 50mm thick that were derived by "removing" defects as shown in Figure 3. The length distribution of blanks therefore depends on the grade of boards. The higher the grade, the less the defects appear on the boards and the larger the proportion of longer blanks. Figures 4 and 5 show the distribution of blanks used in this study. Figure 4 describes the blank length distribution by percent pieces while Figure 5 describes the blank length distribution by volume.

Figure 4. Blank length distribution (percent of pieces)

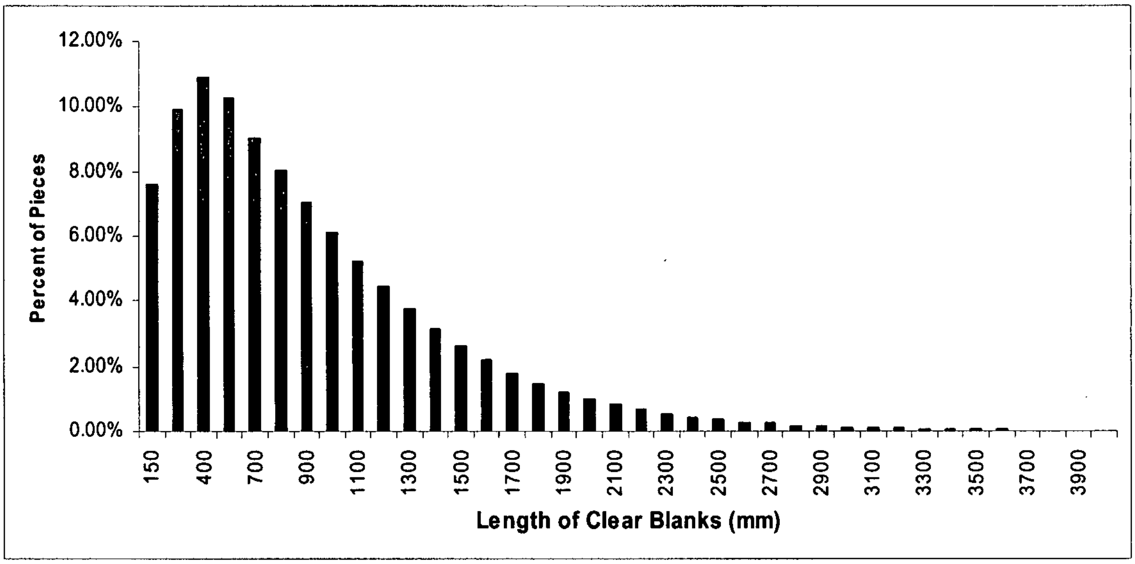
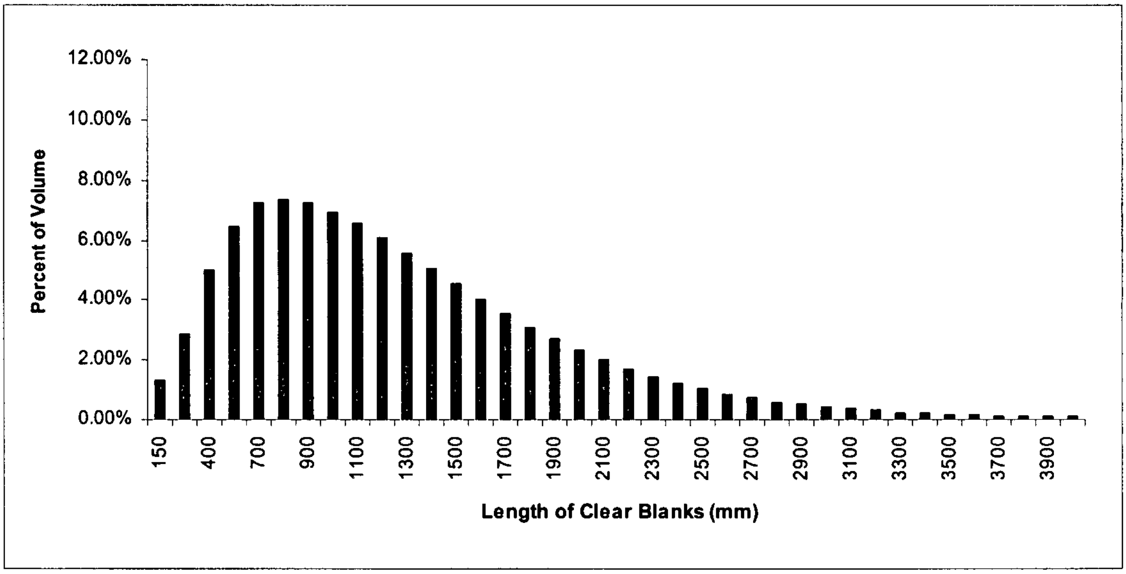


Figure 5. Blank length distribution (percent of volume)



3.2.2 Cutting Lists

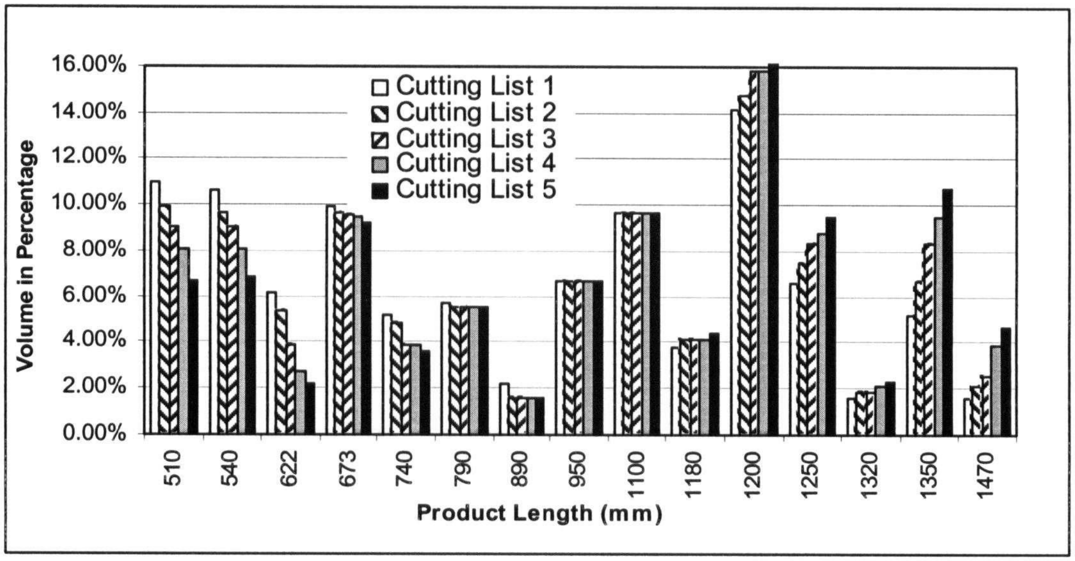
Five cutting lists of varying difficulty were used to test the optimization method described in the next section. The cutting lists shown in Table 1 give the product description, length, order quantity and the value of each part to be produced in a production run. As all the parts have the same width and thickness as the blanks, length is the only dimension variable. In this study, the rough mill is producing parts for their own use, so the "value" is difficult to determine. The mill must simply produce the part list required by a given time at the minimum cost. If a part length combination can be found that exactly fits in the blank (with kerf), then the best part yield is achieved from the blank. Therefore, part length is used as value for each part.

Table 1. Cutting lists for the study

Item Number	Part Length	Value	Required Pieces				
	mm		Cut List 1	Cut List 2	Cut List 3	Cut List 4	Cut List 5
D510	510	510	732	666	606	540	444
D540	540	540	672	612	570	510	432
D622	622	622	336	294	210	150	120
D673	673	673	504	492	486	480	468
D740	740	740	240	222	180	180	168
D790	790	790	246	240	240	240	240
D890	890	890	84	60	60	60	60
D950	950	950	240	240	240	240	240
D1100	1100	1100	300	300	300	300	300
D1180	1180	1180	108	120	120	120	126
D1200	1200	1200	402	420	450	450	468
D1250	1250	1250	180	204	228	240	258
D1320	1320	1320	42	48	48	54	60
D1350	1350	1350	132	168	210	240	270
D1470	1470	1470	36	48	60	90	108

To make model performance on the different cutting list comparable, the five cutting lists have the same parts. The total volume of parts required by each cutting list is also the same. The only difference among the cutting lists is the length distribution of parts required (Figure 6). They were ranked from “easy to get” to “hard to get”. An “easy-to-get” cutting list is one that contains a larger portion of short parts which are easy to cut. A “hard-to-get” cutting list is one that contains a larger portion of long parts. These parts are difficult to cut because the system continues to search for blanks long enough to produce the parts. Cutting list 1 is the easiest because the percentage of long parts is less compared with other cutting lists. Cutting list 5 is the hardest because a larger percentage of long parts is required.

Figure 6. Part length distributions of the five cutting lists



The information on the inventory cost for each part is shown in Table 2. These consist of storage and handling costs, an opportunity cost on the working capital, and the damage cost. These costs depend on the value and the demand of the parts.

Table 2. The cost of overproduction and underproduction for different parts

Item No.	Length (mm)	Overproduction					Underproduction ($/m3)
		Damage	Opportunity Cost	Storage ($/m3)	Handling ($/m3)	Total ($/m3)	
D510	510	20%	1.5%	15	25	140	1600
D540	540	20%	1.5%	15	25	144	1650
D622	622	20%	1.5%	15	25	131	1700
D673	673	20%	1.5%	15	25	131	1750
D740	740	20%	1.5%	15	25	131	1800
D790	790	25%	1.5%	30	25	171	1850
D890	890	25%	1.5%	30	25	209	1900
D950	950	25%	1.5%	15	25	203	1950
D1100	1100	25%	1.5%	15	25	203	2000
D1180	1180	25%	1.5%	30	25	209	2050
D1200	1200	10%	1.5%	15	25	112	2100
D1250	1250	25%	1.5%	15	25	203	2150
D1320	1320	10%	1.5%	15	25	112	2200
D1350	1350	25%	1.5%	15	25	203	2250
D1470	1470	10%	1.5%	15	25	112	2300

3.2.3 Optimization Methods

The following simulation program was developed to compare the performance of the goal-seeking model with Thomas's model.

First, to simulate a real-time process, a random sampling procedure was designed to randomly select a blank from the raw material pool illustrated in Figures 4 and 5. Therefore, the selection of blanks is stochastic, resulting in a slightly different result each time the model is executed.

A dynamic programming (DP) algorithm is used in the model to generate optimal cutting patterns. DP is an optimization procedure that is particularly applicable to chop saw optimization problems, which require a sequence of interrelated decisions. Unlike exhaustive searches which examine every possible solution, DP finds the best solution in an efficient way. The methodology employed in DP is to solve a "whole problem" by solving sub problems beginning at the end of the "whole problem". Due to computational efficiency, the DP algorithm is applicable in a real-time process where the decisions need to be made in less than a second.

A DP algorithm based on the knapsack algorithm is used to determine the optimal cutting patterns for each blank (Dreyfus and Law 1977). The formulation is:

$$f(L) = \max_{y_k \in Y(L)}[v(y_k) + f(L - y_k - c)] \qquad [5]$$

for $0 \le L \le L_{\max}$ and with $f(0) = 0$

where:

$f(L)$ the maximum possible value obtained from chopping a blank of length L;

$Y(L)$ the set of feasible parts at L for all k parts;

$v(y_k)$ the value of part y_k;

$L_{\max}$ maximum blank length

c the width of a kerf for each cut. In this study, $c = 5mm$

Kerf loss is considered in the model at the rate of 5mm for each cut.

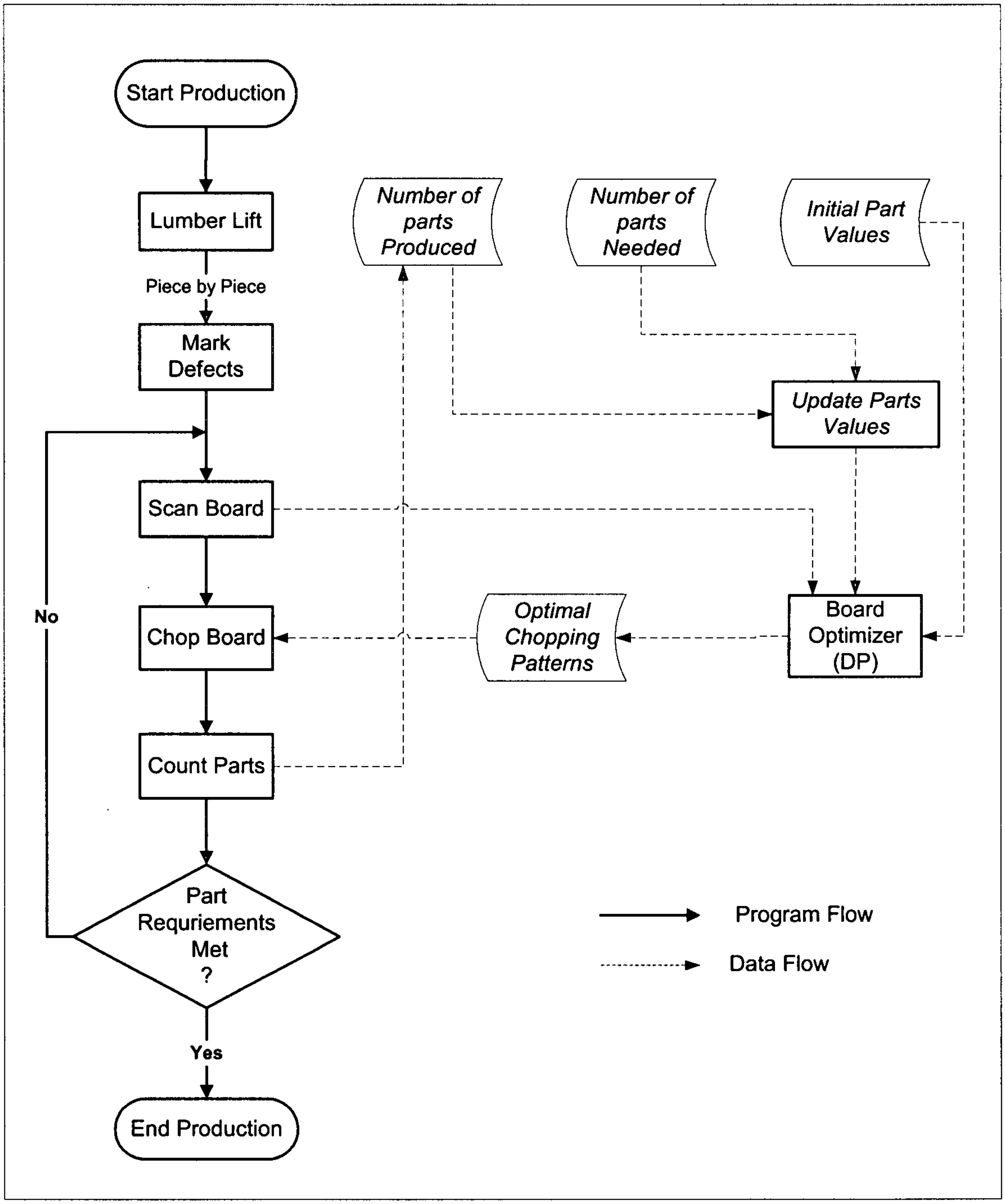
With this DP problem, the optimal cutting pattern for each blank of length L can be calculated in an efficient way.

As described above, in a typical first generation board optimizer the part values $v(y_k)$ are static throughout the production run. This method yields the maximum value if the chop saw can produce any quantity of each part, in other words, the costs of over-production and under-production are zero. However, this is rarely true in practice. To make the model practical, $v(y_k)$ must be dynamically updated to adjust the production of part y_k in order to meet the cutting list requirement.

The goal-seeking model is based on a two level structure. On the first level there is a blank-by-blank optimization algorithm with the objective to maximize the part value combination from each blank. On the second level there is a process control procedure which serves to look at the quantity requirement for each part on the cutting list and the quantity produced for each part during the production process, and dynamically adjusts $v(y_k)$ for each part on the cutting list each time a blank is chopped.

The flow chart for the Goal-seeking model is demonstrated in Figure 7. First a grader marks the positions of the defects on an incoming board with reflective crayon. The board is then sent to a scanning system where the positions of the defects are scanned to determine the blank length of each defect-free portion in the board. The "Optimizer" in the chop saw collects the information from both the scanning system and the cutting list and then calculates the cutting pattern for the blank detected. Any blank less than the shortest part length, in this case, 510mm, becomes waste. The chopping instructions are then sent to the saw unit to chop the blank into different parts. A counter, located right after the chop saw records the number of each part produced and sends the information back to the process control procedure in the system. The process control procedure then updates $v(y_k)$ based on the number of parts produced and the number of parts needed on the cutting list each time a blank is chopped. The updated $v(y_k)$ will be used by DP to generate a new cutting pattern for the next blank. This loop is carried throughout the production process. Once the quantity produced for a part reaches the corresponding quantity required on the cutting list, the corresponding $v(y_k)$ will turn to a

Figure 7. Program Flow Chart of Goal-seeking model



very small number "$s(y_k)$" which is larger than the value of waste. In this way, the over-production is allowed to maintain appropriate part yields. The production continues until all the requirements on the cutting list are met.

Two prioritizing strategies are employed in the process control procedure: Simple Dynamic Value (SDV) and Complex Dynamic Value (CDV).

The SDV strategy uses part length as initial part value and the prioritization is:

$$v(y_k) = \frac{Qty'_{y_k} - Qty_{yk}}{Qty'_{y_k}} \times Length \qquad [6]$$

where:

Qty'_{y_k} Quantity initially required for part y_k in the cutting list

Qty_{y_k} Quantity produced for part y_k so far in this production run

Length Initial part length

The CDV strategy increases the priority of long parts by using squared part length as initial part value and the prioritization is:

$$v(y_k) = \frac{Qty'_{y_i} - Qty_{y_i}}{Qty'_{y_i}} \times Length^2 \qquad [7]$$

With these prioritizing strategies, the part value $v(y_k)$ can be adjusted each time a blank is chopped.

Equation [6] and [7] are nominally based on fuzzy set theory, in which the value of the part is associated with the degree of membership in a fully produced order. Therefore, the goal is a fully filled order, and the dynamically changing part values seek the goal. The length squared term gives a higher priority to longer parts to reflect that these are more difficult to produce. In this way, the part values on the cutting list are linked with the levels of production for each part, and each time a blank is chopped, the part value table is updated. These values reflect the current state of production. Therefore, the board optimizer is always using part values that reflect the latest information and adapts to the changes that are occurring on the plant floor.

3.3 RESULTS AND DISCUSSION

This section compares the performance of the Goal-seeking model with Thomas's model. Four prioritization strategies were simulated:

SDV and CDV developed in this study and SDE and CDE developed by Thomas (1996).

As the sampling procedure is a stochastic process, the simulation result is slightly different for each run. Therefore, the results shown below were obtained by taking the average from five runs for each strategy, against each cutting list. The figures indicate the averages and the one standard deviation intervals.

3.3.1 Yield Performance

Yield is the ratio of total volume of parts required on a cutting list to all the raw materials required, including the raw materials expended for overproduced parts.

A higher yield means that fewer raw materials are needed to produce the cutting list.

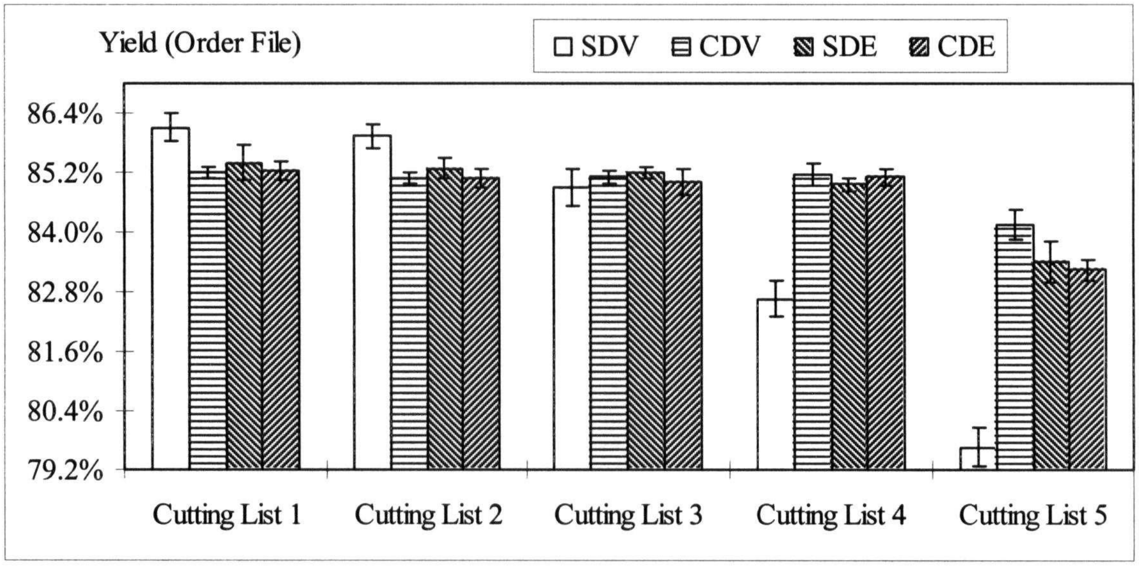
Figure 8 shows the performance of the two models with different prioritization strategies against the five cutting lists.

In Figure 8, of all the four prioritization strategies, SDV achieves the highest yield on the easy-to-get cutting lists (cutting list 1 and 2). In the simulation process all the part on the cutting list were produced simultaneously until they were finished. This allows the optimization program to easily find the optimal chopping patterns as there are maximum choices for the program to choose. Therefore, the highest yields are achieved. This conforms to the result found by Maness and Wong (2002).

However, as the cutting list gets harder the yield performance for SDV drops significantly. The worst performance can be observed for the SDV strategy on the hard-to-get cutting lists (cutting list 4 and 5). The problem with the hard-to-get cutting lists is that near the end of the production run all the short parts and medium parts on the cutting list are satisfied, leaving some of the long parts incomplete. As the process continues to search for long boards to produce the parts remaining, excessive overproduction occurs and the yield drops.

In contrast to SDV, CDV achieves a lower yield for easy-to-get cutting lists and a higher yield for hard-to-get cutting lists. For example, the yield from CDV against cutting list 5 is 84.15%, higher than the other three strategies. This indicates that using CDV to produce the hard-to-get cutting list will obviously reduce the amount of boards used and the amount of waste generated. This increases the efficiency of production because fewer boards have to be processed for the same cutting list, and this subsequently leads to the reduction of the overall cost for the production run.

Figure 8. Order file yield – ratio of total volume of parts required on a cutting list to all the raw materials used

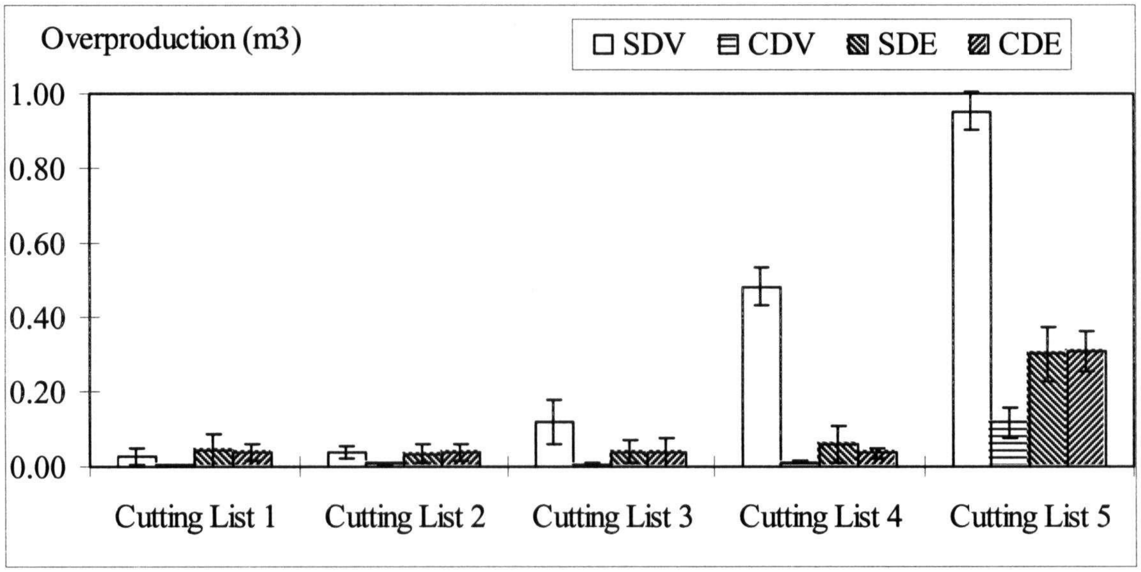


3.3.2 Over-Production Performance

The information on the amount of over-production is recorded by running the model with four prioritization strategies against the five cutting lists.

From the result shown in Figure 9 we can find that SDV performed well on the easy-to-get cutting lists (cutting list 1 and 2), but not for hard-to-get cutting lists (especially cutting lists 4 and 5). This explains its poor yield performance against the hard-to-get cutting lists (Figure 8).

Figure 9. Volume overproduced by different prioritization strategies against the five cutting lists



Of all the four prioritization strategies, CDV has the best performance on all the five cutting lists. Although SDE and CDE strategies have good performances on cutting list 1 to cutting list 4, they do not perform well on cutting list 5, which is the hardest one to get in this study.

These results indicate that if the inventory cost is high, the SDV, SDE and CDE strategies can cause a serious loss in profit for the rough mill.

3.3.3 Overall Cost Performance

The overall cost for filling a given order consists of the cost of raw material (boards), the penalty for over-production of each part as shown in Table 2 and the cost for disposing the waste. Among these costs, the raw material cost and the cost of over-production are dominant. A cost of 400 $\$/m^3$ is charged for the raw material and 50 $\$/m^3$ for the dumping of waste in this study.

The outcomes for running the model with each prioritization strategy against the five cutting lists are shown in Table 3 and Figure 10.

Table 3. Overall cost performance of the goal-seeking model (SDV and CDV) and Thomas's model (SDE and CDE)

		Overall Cost ($)				
Model	Strategy	Cutting List 1	Cutting List 2	Cutting List 3	Cutting List 4	Cutting List 5
Goal-seeking Model	SDV	$8,062	$8,080	$8,197	$8,474	$8,865
	CDV	$8,156	$8,167	$8,163	$8,160	$8,276
Thomas Model	SDE	$8,139	$8,150	$8,157	$8,186	$8,440
	CDE	$8,155	$8,170	$8,178	$8,168	$8,402

Figure 10. Overall cost by different prioritization strategies against the five cutting lists

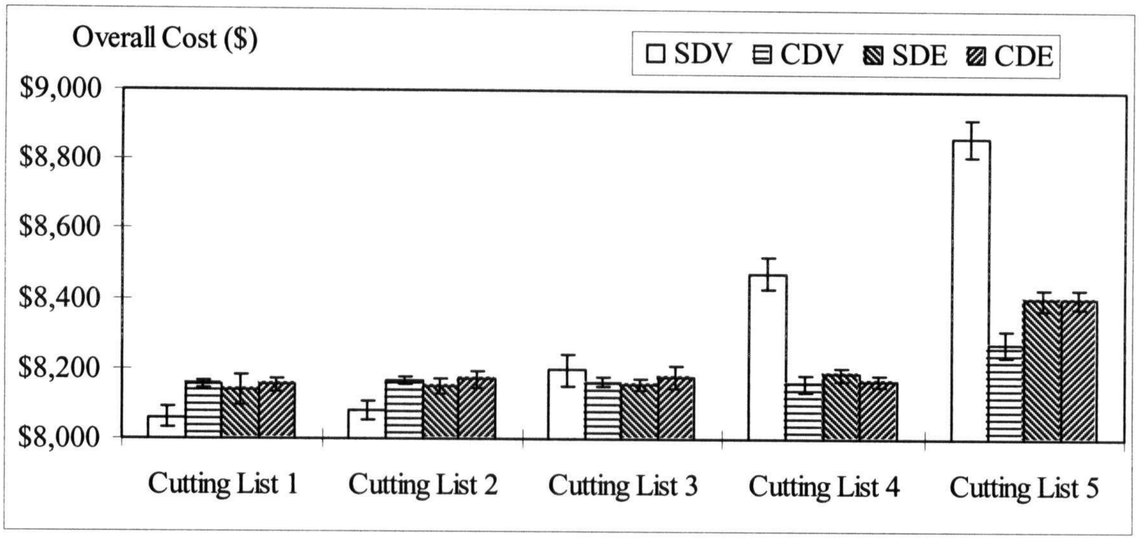


Table 3 and Figure 10 show that SDV strategy got the lowest overall cost from cutting list 1 and 2. This is due to the higher yield shown in Figure 2 and low level of overproduction shown in Figure 3.

Table 3 and Figure 10 also show that the CDV strategy got the lowest overall cost from cutting list 4 and 5. The tighter control of the overproduction by the CDV strategy contributes the most to the cost performance because less inventory penalty is applied to the overall cost.

3.3.4 Production Progress Performance

An example of the production progress with the goal-seeking model is shown in Figure 11 to Figure 13. It can be seen in the Figures that production using the goal-seeking model progressed at a balanced rate for each part. This is because the part values on the cutting list for goal-seeking model are linked with the production level for each part and updated each time a blank is chopped. If a part is chopped from a blank at this time, higher priority will be placed on other parts for the next blank

Figure 11. Production progress of the goal-seeking model (at 25% of total production run)

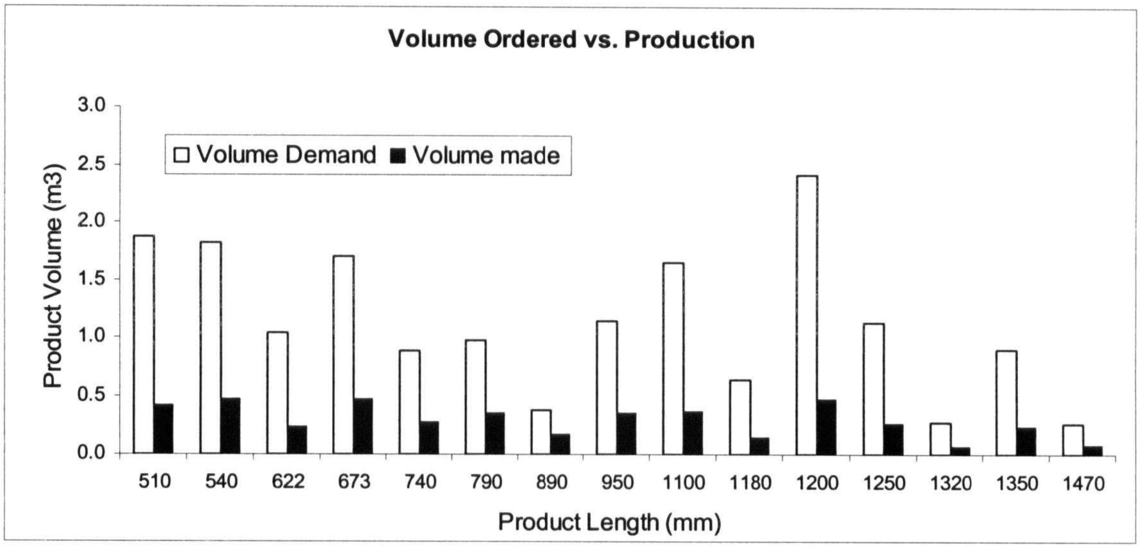


Figure 12. Production progress of the goal-seeking model (at 50% of total production run)

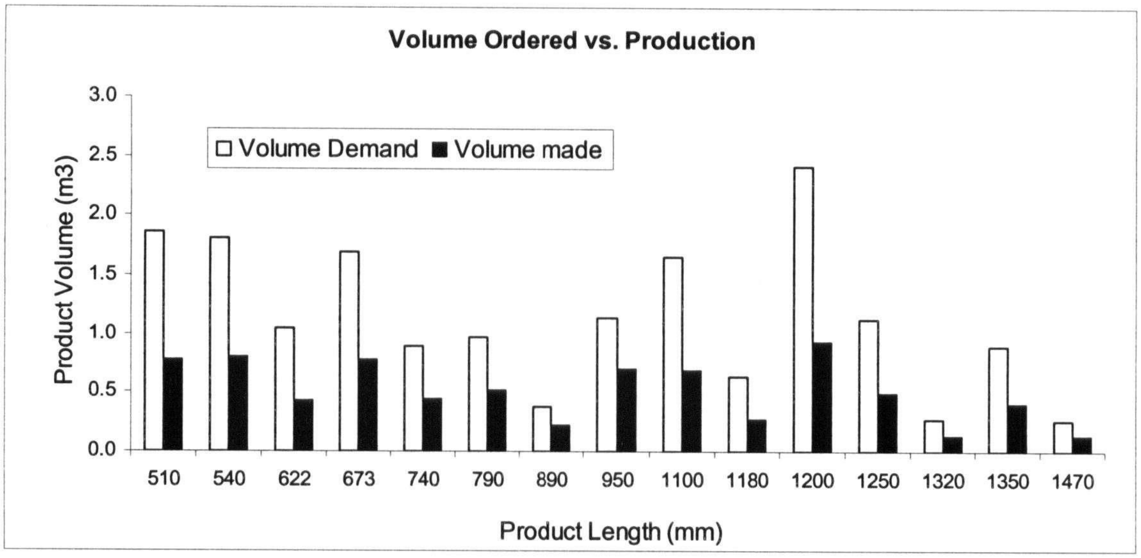
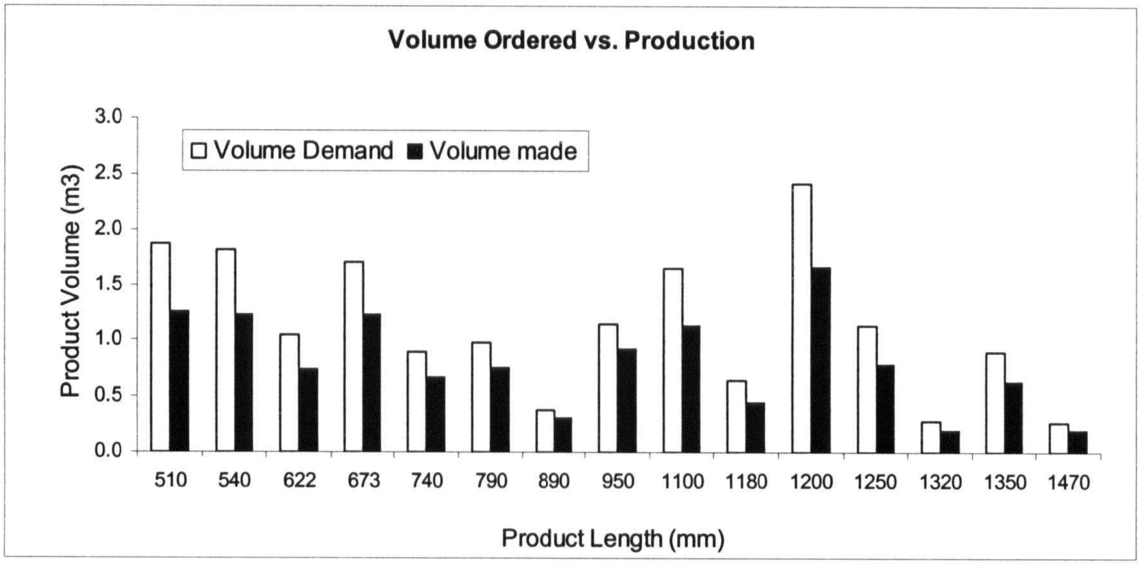


Figure 13. Production progress of the goal-seeking model (at 75% of total production run)



3.4 SUMMARY

Producing quality product to meet orders at the least cost continues to be a challenge for operating chop saw systems. The purpose of this study was to develop a new model for chop saw optimization systems that can be implemented on-line and can cut boards consistently with higher yield, less overproduction and ultimately at the least overall cost.

The study found that goal-seeking model with the SDV strategy has the best performance for easy-to-get cutting lists while the model with the CDV strategy is well suited to hard-to-get cutting lists. If SDV and CDV were incorporated into the on-line chop saw optimization program, significant saving of overall cost could be achieved.

As a real-time decision-making tool, the model produces just those parts and quantities required on the cutting list. This advantage becomes more important when the cost for the inventory is high. The model will continually balance what already has been cut with the remaining requirements for the production run. At the end of the run, the different products required are produced in a balanced ratio.

CHAPTER 4

4.0 COMBINED MODEL

4.1 INTRODUCTION

Linear programming (LP) is a good tool for minimizing the overall cost over a production run, but it is not good at making decisions in real-time. Dynamic programming (DP) is very good at optimizing the cutting solution of individual blanks but is not suitable for minimizing the cost of the entire production run. This chapter describes an economic approach that combines LP and DP to solve the problem defined in "Definition 4". Adaptive methods are used to get the knowledge of the raw material distribution and the level of production for each part in real time. The objective of the model is to minimize the overall cost of producing a cutting list for a chop saw production run. A study was conducted to compare the combined model with the Goal-Seeking Model described in Chapter 3.

4.2 MATERIALS AND METHODS

4.2.1 Raw Materials

The raw materials used in this study are identical to the raw materials used in Chapter 3. The distribution of blank lengths is shown in Figures 4 and 5.

3.2.2 Cutting Lists

The same cutting lists in Table 1 were used in this study. The information on the cost of overproduction and underproduction are also the same and are shown in Table 2.

The cost of disposing waste is $50 per cubic meter.

4.2.3 Optimization Methods

4.2.3.1 The Solution Approach

The proposed optimization model for the chop saw system is a multi-criteria optimization problem that considers the following:

1. raw material cost,
2. sawing technology,
3. order requirement (cutting list),
4. cost of holding inventory, dunnage and material handling cost,
5. cost of delivery delay, and
6. cost of disposing the waste.

Linear programming is a well-known technique that is used to minimize costs or maximize profits by allocating limited resources to different activities in an optimal manner. If LP was used to solve the chop saw optimization problem, the constraints of the model could serve to ensure that the order files are met, the sawing technology is enforced, and the problem is balanced from a material usage standpoint.

Formulated in this way, the key activities of the LP with respect to optimization would be the chopping patterns for each blank length. Simply put, the solution of the LP model would indicate the optimal ways to chop each blank into parts as required by the cutting list to minimize the overall cost.

To make the linear programming applicable to the real-time chop saw problem, the following factors were considered:

1. Problem size. When many different parts are produced in a plant there are a large number of possible chopping patterns for each blank length. To minimize the cost, all possible chopping patterns should be considered in the LP model. This results in a huge LP model that may be unsolvable in real time. If the number of chopping patterns used in the LP model for each blank is only a sub set of all possible chopping patterns, the solution of the model may be sub-optimal.
2. Distribution of blank lengths unknown. To fully optimize the production run, the system needs to know the distribution of blank lengths before optimization. However, in a typical rough mill each board is scanned just prior to chopping. The full distribution is not known until all of the boards have been chopped, and by this time it is too late to optimize.
3. Split sawing solutions. The LP model, having an assortment of chopping patterns for each blank to choose from, will often chop a given blank with more than one pattern. These split decisions are impossible to implement with a chop saw system as only one of the chopping patterns can be selected for a given blank.

To solve the first problem (problem size), a combined LP-DP model was formulated using the Column Generation Technique (Dantzig & Wolfe 1960, Maness & Adams 1991). The technique was first used to solve the cutting stock problem in which the profit of cutting a set of raw materials into a set of products of different dimensions and quantity requirements is maximized. It divides the cutting stock problem into two problems: a master problem and a sub-problem.

The master problem in the combined model is formulated as a LP, with minimizing overall cost as the objective and the available raw material, production technology, and product demands from the cutting list as the constraints. The key activities with respect to optimization of the LP are the selection of chopping patterns that produce the order at minimum cost. The sub-problem is a DP algorithm that suggests optimal chopping patterns based on the value of each part, in this case formulated as a knapsack algorithm (Dreyfus & Law 1977). The two models are connected as a column generation LP (Dantzig & Wolfe 1960).

The connections between LP and DP are the Lagrangian multipliers (shadow prices) associated with the product demand constraints in the LP, and the optimal chopping patterns generated by DP. The shadow prices generated by the LP serve to update the part value table for the DP. Therefore, the chopping patterns generated by the DP using these shadow prices will produce parts that most decrease the cost (overall cost) of the objective function of the master problem. These chopping patterns are then added to the LP as additional columns that the LP can choose from. This reduces the total number of columns in the LP tremendously, as only the best possible patterns are selected by the sub-model to be considered for inclusion in the LP.

The model is executed iteratively until there are no new chopping patterns generated by the sub-problem. The final solution of the LP indicates which chopping pattern should be used for an incoming blank to minimize the overall cost.

However, the chop saw cutting optimization is far more complicated than the cutting stock problem, due to the second problem described above, namely that the true blank length distribution is unknown at the start of the production. To solve the second problem, an adaptive method was used. During the production run, each board is scanned and the number of blanks for each length class is recorded. Over the course of the production run, the distribution of blanks derived from the boards is built up from zero to complete knowledge at the end of the run. Therefore, the LP must be run at intervals during the production run as knowledge about the raw material builds. Each time we solve the optimization problem the current accumulated blank length distribution is used as an estimate of the entire blank length distribution. Therefore, the raw material distribution in the LP matrix always uses the latest information collected from the scanning unit as an estimate of the entire raw material. This adaptive method will be described in detail later. In this example multiple grades (or quality) of blanks are not needed but this could be added to the method without loss of generality.

A similar adaptive method should also apply to the part requirement as it changes dynamically along with the production process. Each time the LP solves the problem, the part requirement data in the LP matrix is updated to reflect the latest information collected on the plant floor.

To solve the third problem (split sawing solutions), a probability-based sampling procedure was used to select one of the chopping patterns if the LP solution contains split chopping decisions for a particular blank length. The probability of selecting a given chopping pattern is proportional to its corresponding activity level in the LP. In other words, the larger the amount of activity in LP for a certain chopping pattern in a set of split chopping solutions, the higher the probability this chopping pattern will be selected in the real-time system. Therefore, each time the real time chopping program queries the LP for a solution it will get one chopping solution even if there are multiple solutions in the LP for a particular blank length. When using this method over the course of a production run, the split decisions are generally chosen in the proportion close to the optimal.

4.2.3.2 Formulation of the Combined Model

The combined model consists of a master LP for plant optimization and a DP-based sub-model for chopping pattern optimization. Each of the models is described below.

4.2.3.2.1 The Master LP

The Master LP consists of an objective function and a series of constraints. The objective of the model is to minimize the overall cost for the plant. There are 3 sets of constraints: raw material constraints, production technology constraints, and product demand constraints. The formulation is as follows[1]:

Minimize:

$$\begin{aligned}&\mathit{LumberVol}\cdot\text{LumberPrice}+\mathit{FiberVol}\cdot\text{DispCost}\\&+\sum_{k=1}^{m}(\mathit{OverProd}_k\cdot\text{InvCost}_k+\mathit{UnderProd}_k\cdot\text{UnderCost}_k)\end{aligned}\qquad[8]$$

Subject to:

<u>Material Balance Constraints</u>

$$\sum_{i=1}^{n}(\mathit{LumberVol}\cdot\text{BlkDist}_i)-\sum_{i=1}^{n}\sum_{s=1}^{t}\mathit{ChopPat}_{is}=0\qquad \text{v}\ i\qquad[9]$$

<u>Product Recovery Constraints</u>

[1] All variables hereafter are written in italics and coefficients in non-italics.

$$\sum_{i=1}^{n}\sum_{s=1}^{t}(ChopPat_{is}\cdot\mathrm{Recovery}_{isk}) - TotProd_{k} = 0 \qquad \vee\, k \qquad [10]$$

$$\sum_{i=1}^{n}\sum_{s=1}^{t}(ChopPat_{is}\cdot\mathrm{RecoveryFiber}_{is}) - Fiber\,Vol = 0 \qquad [11]$$

Product Demand Constraints

$$TotProd_{k} - OverProd_{k} + UnderProd_{k} = ReqProd_{k} \qquad \vee\, k \qquad [12]$$

where i=1, 2, …n; k=1, 2, …m; s=1, 2 …t.

Subscript definition: i=lengths of blanks; k=parts; s=iteration number.

Definitions:

$\mathrm{BlkDist}_i$ the percentage of clear blank i from LiftVol.

$\mathrm{ChopPat}_{is}$ the volume of clear blank i from iteration s.

DispCost cost of disposing a unit of fiber.

FiberVol fiber (waste) volume in total.

$\mathrm{InvCost}_k$ per unit inventory cost for part k.

LiftPrice lift price.

LiftVol lift volume.

$\mathrm{OverProd}_k$ the volume of over-produced part k.

$\mathrm{Recovery}_{isk}$ the percentage of clear blank i from pattern created in iteration s assigned to part k.

$\mathrm{RecoveryFiber}_{is}$ the percentage of clear blank i from iteration s assigned to fiber (waste).

ReqProd_k	the volume of part k required by the cutting list.
TotProd_k	the total volume of part k produced.
UnderCost_k	per unit cost of buying part k from alternative sources.
UnderProd_k	the volume of under-produced part k.

Objective function

The objective function in equation [8] is a cost minimization formulation. The costs are divided into four categories: raw material cost, cost of disposing the waste, cost of holding inventory for over-produced parts, and replacement cost for under-produced parts.

Material Balance Constraints

Equation [9] is a raw material balance constraint that ensures that the volume of lumber that the plant processes does not exceed what is available. Equation [9] also allows each blank to be assigned to one or more chopping patterns. A lift is a package of boards that are the same dimension length and grade, and are packed together and shipped to the plant by truck or by train. Boards from a lift are sent, piece by piece, to the scanning system to identify the location of marks which distinguish between defects and blanks. The raw material distribution used by this model is the distribution of blank lengths that are obtained after the defects have been identified by the graders and marked out with a grading crayon. Complexity can be added to the basic model by adding thickness, width and grade to the formulation.

Product Recovery Constraints

There is one constraint for every raw material class (blank length) in the model. The constraint contains the technical coefficients (Recovery$_{kjp}$) which chop each blank into its respective parts for that chopping pattern. These technical coefficients are organized into columns, one column in the LP for each possible chopping pattern in the model. These chopping patterns are generated by the dynamic programming sub-model.

Product Demand Constraints

These are formulated as soft constraints that allow both over- and under-production. There is an inventory cost for overproduction and a replacement cost for underproduction. Associated with these constraints are the shadow prices of the products used by the DP.

Solving the LP

The solution to this LP problem gives the set of chopping patterns that are applied to each blank to meet production orders and to achieve the minimum overall cost. As described above, one problem with solving this as a straight linear programming model is that the full set of possible chopping patterns makes a very large LP. Therefore, the column generation approach is used to reduce the size of the model and decrease the execution time.

4.2.3.2.2 The DP model

The DP used in this model is based on the Knapsack algorithm (Dreyfus & Law 1977). The DP model serves to look at each possible blank length and generate the current optimal chopping pattern columns for the LP, aiming to maximize the part value from each blank. Each time the DP is executed it adds a new set of columns to the LP. The formulation of the model is:

$$f(L) = \max_{y_k \in Y(L)}[v(y_k) + f(L - y_k - c)] \qquad [13]$$

for $0 \le L \le L_{\max}$ and with $f(0) = 0$

where:

f(L) the maximum possible value obtained from chopping a blank of length L;

Y(L) the set of feasible parts at L for all k end-use parts;

v(y_k) the value of part y_k;

L_{max} maximum board length;

c the width of the kerf for each cut. In this study, c=5mm.

With this recursive dynamic programming algorithm, the optimum chopping pattern for each blank length can be easily found. In a typical first-generation chop saw optimization program the part values *v(y_k)* are static throughout the production run. In second-generation models the part values typically have overriding priorities that force certain priority parts into the solution (Chow 1998). However, in this proposed third-generation combined model, the part values are the link between the Master LP and the DP. Each time the combined model is executed it provides the current set of part "opportunity costs" associated with the product constraints in the LP. These values reflect the current knowledge about the raw material distribution and the current state of production levels for each part.

4.2.3.3 General Description of the Combined Model

The flow chart of the Real-Time combined model is shown in Figure 11. The master problem is a cost-minimizing LP and it models the chop saw line as a cost center in the plant. The sub-problem is a dynamic programming model based on the Knapsack algorithm that maximizes the part value from each blank length.

At the beginning of a production run, the LP in the combined model is initialized by loading two initial data files to its matrix. The first data file is the initial distribution of blank lengths derived from the previous knowledge on the quality of the raw material. The second data file is the part quantity requirements from the cutting list. The initial part value data file is loaded to the DP in the combined model as well so that the DP can be initialized to generate a set of chopping patterns, one for each blank length. These chopping patterns are subsequently added to the LP matrix as columns. The LP is then executed to calculate the shadow prices associated with the product demand constraints in the LP to update the part values in the part value data file. These new values represent the current best estimate of the marginal value of each of the parts to the plant and they are used by DP to generate new chopping pattern columns for the LP. This iterative process is executed until there are no new chopping pattern columns generated by the DP. The final solution of the LP indicates which chopping patterns should be used on which blank to minimize the overall cost for the production run.

Although the iterative process increases the number of columns in the LP as the DP continuously adds new columns (chopping patterns), the computing time of the LP decreases significantly after the first iteration[2]. This is due to the fact that the solver used for the LP problem is running continuously during the iterative process. Consequently, after an optimal solution is found, the solver saves the matrix of the LP problem and keeps the list of the basic variables in the computer memory. In each iteration, only the data for the new columns is loaded into the solver. Moreover, after new columns are added to the LP problem, the solver finds an optimal solution by evaluating the potential of the new columns to enter the basis saved during the previous iteration, instead of solving the whole problem from the very beginning. Therefore, as the number of new columns generated in each iteration decreases, computational time also decreases.

During the production process, the quantity of each blank length observed is recorded by the scanning system. Consequently, the knowledge of the raw material distribution is constantly being improved. An estimate is always available to periodically update the blank length distribution data in the LP matrix. Simultaneously, the quantity of each part produced is also recorded by a scanner located right after the chop saw so that the part requirements data file and subsequently the right hand side of the product demand constraints in the LP matrix are periodically updated during the production run.

The blank length distribution and the part demands in the LP are updated whenever the number of blanks processed reaches a specified number N. Then the combined optimization model is executed again, now using the new blank length distribution based on the cumulative knowledge on the blanks that have already been processed and the updated right hand side of product demand constraints in the LP

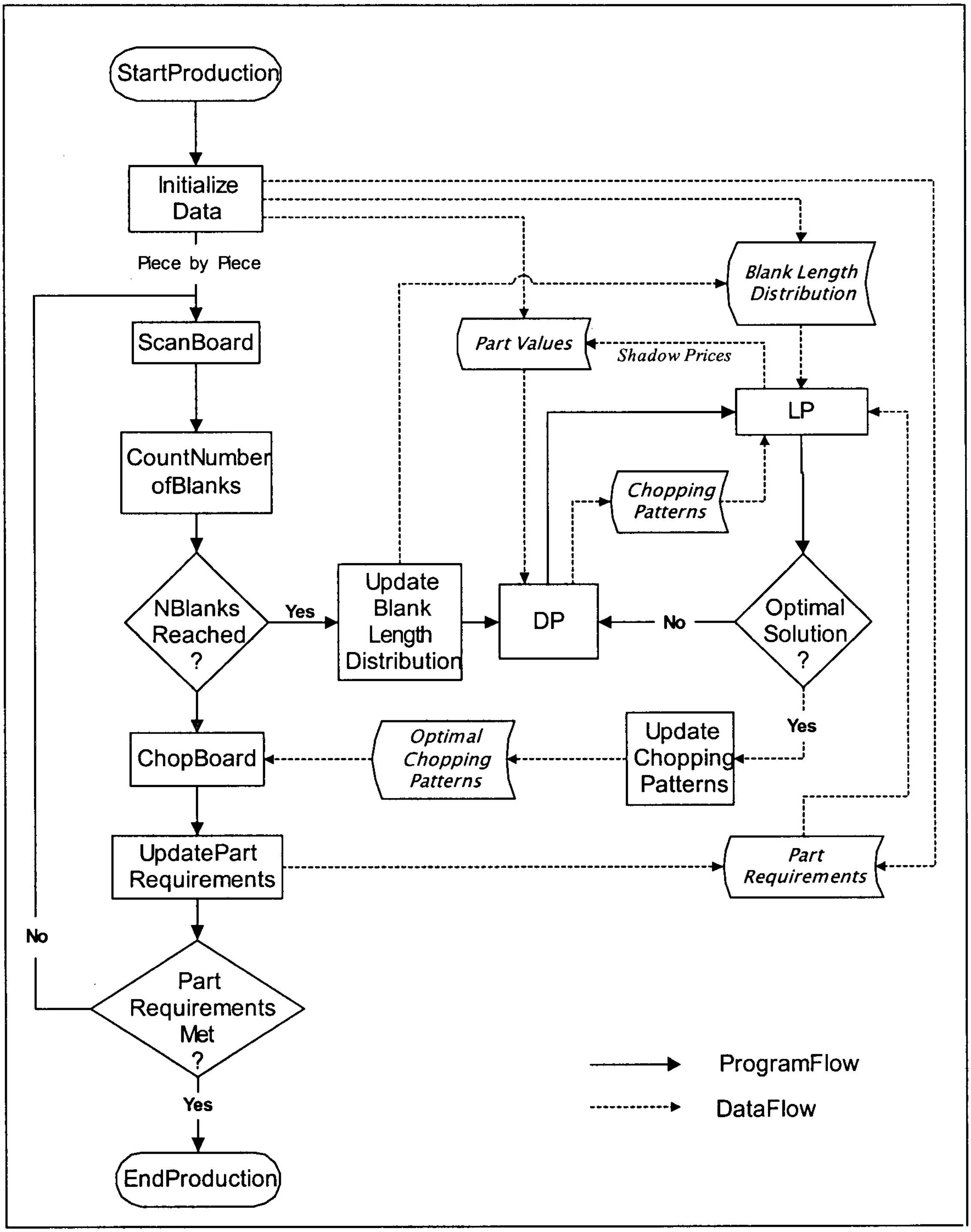
[2] See Hillier & Lieberman (2001), P212-213 for discussion.

matrix. A new set of optimal chopping patterns is found for the next N blanks. This loop is carried out through the entire production run.

With this technique the LP always supplies the optimal chopping patterns to the chop saw to produce the parts required by the cutting list based on the actual board quality and the part inventory levels collected during the production run. The combined model adapts to the changes that are occurring on the plant floor.

The number N in this model represents the solution (chopping pattern) updating frequency. The smaller the number N, the more often the solution is updated, and the better the optimization results.

Figure 14. Flow chart of the combined model using adaptive methods



4.3 RESULTS AND DISCUSSION

A case study was conducted to demonstrate the performance of the combined model using the adaptive methods. Three different updating frequencies when N=25, N=15, N=5 were simulated in the combined model to study their impact on the results.

The performance of the model was compared to the performance of the goal-seeking model with two prioritization strategies (SDV and CDV) described in Chapter 3.

Overproduction, yield and ultimately the overall cost were used as performance indicators to compare the two models.

The results shown below were obtained by taking the average from two runs against each cutting list. The figures indicate the averages and the one standard deviation intervals.

4.3.1 Yield Performance

Yield, as described in Chapter 3, is the ratio of the total volume of parts required on a cutting list to all the raw materials, including the raw materials expended for overproduced parts, used to fill the cutting list.

The result of yield obtained by running the combined model with three updating frequencies against the five cutting lists is shown in Figure 15. This indicates that updating frequency plays an important role yield performance. As updating frequency increases, yield performance improves. Higher updating frequency (N=5) can achieve a higher yield from all the five cutting lists than with lower updating frequencies (N=15, N=25).

Figure 15. Yield performance of the combined model with different updating frequencies

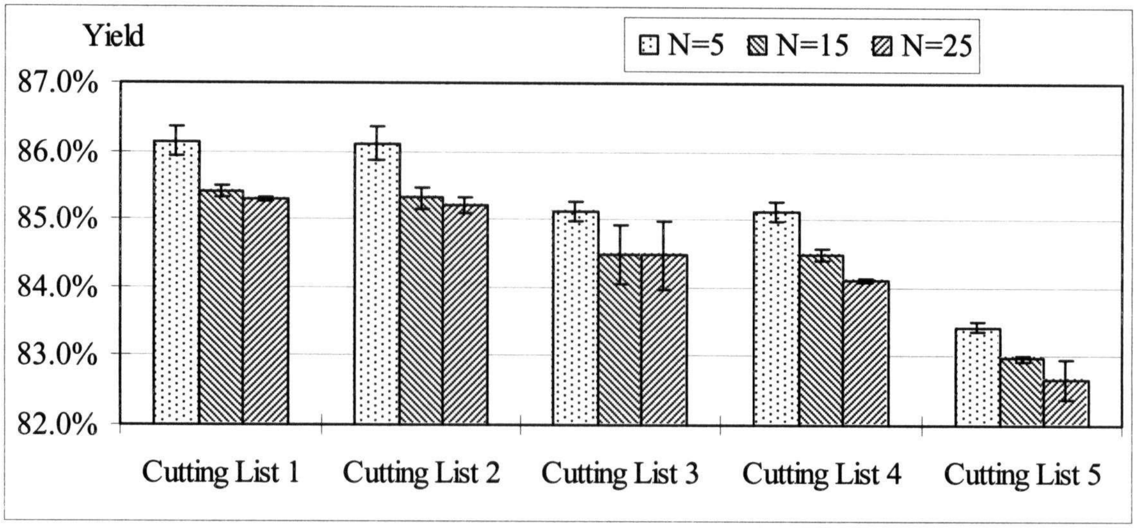
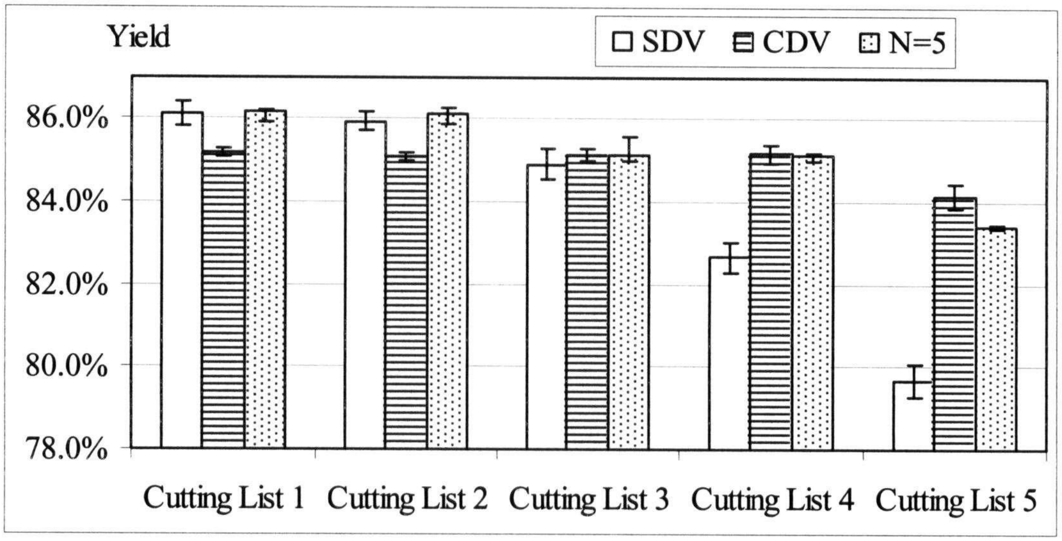


Figure 16 shows the comparison of the yield performances of the goal-seeking model with the combined model when N=5. It can be seen in Figure 16 that the combined model has good performance on cutting lists 1 to 4. For cutting list 5, the combined model achieved a yield of 83.43%, slightly lower than the yield by goal-seeking model with CDV strategy.

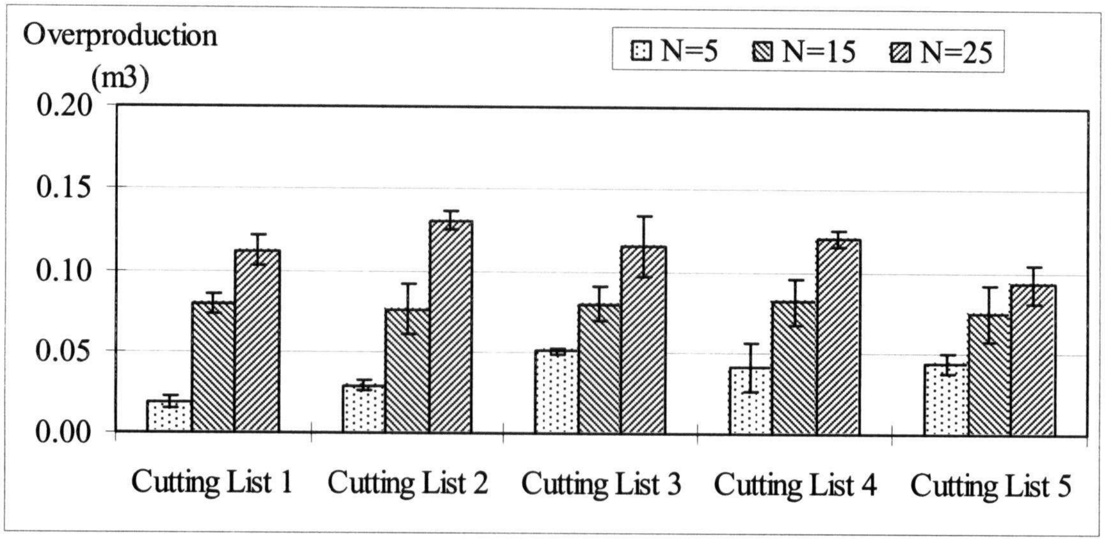
Figure 16. Comparison of the yield performances of the goal-seeking model (SDV and CDV) with the combined model (N=5)



4.3.2 Overproduction performance

The overproduction performance of the combined model with different updating frequencies against the five cutting lists is illustrated in Figure 17. It is obvious that higher updating frequency yields a better result. N=5 is the highest updating frequency in this study and it yields the lowest overproduction. It is also interesting to note that the amount of overproduction stays static from cutting list 1 to cutting list 5 for all the three updating frequencies. For example, the amount of overproduction when N=5 ranges from $0.02m^3$ to $0.04m^3$. This shows that the combined model has a robust control over the amount of overproduction across different cutting lists.

Figure 17. The overproduction performance of the combined model with different updating frequencies against the five cutting lists



The comparison of the combined model when N=5 and the goal-seeking model is shown in Table 4.

Table 4. Comparison of the overproduction performances of the goal-seeking model with the combined model when N=5

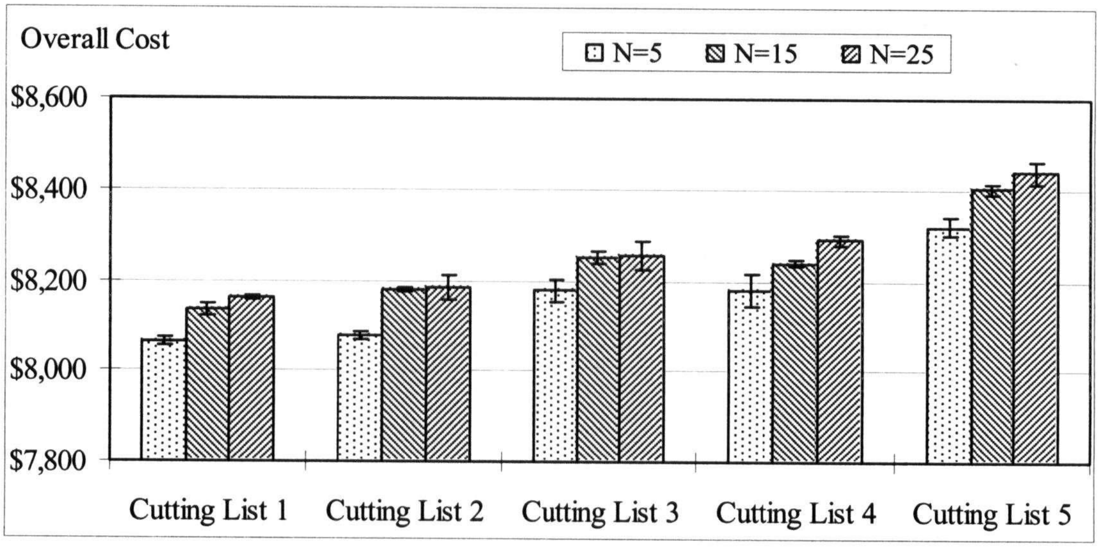
	Overproduction (m3)				
	Cutting List 1	Cutting List 2	Cutting List 3	Cutting List 4	Cutting List 5
SDV	0.03	0.04	0.12	0.48	0.95
CDV	0.00	0.01	0.00	0.01	0.12
N=5	0.02	0.03	0.05	0.04	0.04

Table 4 shows that the amount of overproduction against the five cutting lists by the combined model when N=5 ranges from 0.02m^3 to 0.05m^3. Compared to corresponding results by goal-seeking model with CDV strategy, the combined model has better control of overproduction across different cutting lists. For cutting list 5, which is the hardest one to get, the amount of overproduction by the combined model is 0.04m^3, which is much lower than that achieved by the goal-seeking model with CDV strategy (0.95m^3).

4.3.3 Cost Performance

The cost performance of the combined model with different updating frequencies as illustrated in Figure 18 reveals that higher updating frequency yields a better result.

Figure 18. The cost performance of the combined model with different updating frequencies against the five cutting lists



The comparison of the goal-seeking model with the combined model when N=5 is shown in Table 5.

Table 5. Comparison of the overall cost performances of the goal-seeking model with the combined model when N=5

	Overall Cost ($)				
	Cutting List 1	Cutting List 2	Cutting List 3	Cutting List 4	Cutting List 5
SDV	$8,062	$8,080	$8,197	$8,474	$8,865
CDV	$8,156	$8,167	$8,163	$8,160	$8,298
N=5	$8,066	$8,078	$8,170	$8,180	$8,320

As described in Chapter 3, SDV strategy has good performance for the easy-to-get cutting lists (cutting lists 1 and 2). For hard-to-get cutting lists (cutting lists 3, 4 and 5), CDV strategy performs better. There is no single prioritization strategy that performs well across all cutting lists. However, it is interesting in Table 5 that the combined model when N=5 has the same performance as SDV for easy-to-get cutting lists while for the hard-to-get cutting lists, the performance of the combined model is almost the same as CDV, although slightly higher results than CDV were observed by the combined model.

4.4 SUMMARY

Updating frequency plays an important role to the performance of the combined model. Higher updating frequency achieves a higher yield, lower overproduction and ultimately lower overall cost.

The results of the overproduction by the two models indicate that the combined model can effectively control the amount of overproduction for the five cutting lists, especially the hard-to-get cutting lists used in the case study. This is in contrast to the results of the goal-seeking model where control on the amount of overproduction is only effective on the "easy-to-get" cutting lists. It is also interesting to note that in Figure 17, as the updating frequency increases, the amount of overproduction reduces significantly. This reveals that the updating frequency in the combined model plays an important role in reducing overproduction.

The yield performance of the two models indicates that hard-to-get cutting lists require more raw materials to fill the orders. This is because the harder cutting lists contain a larger proportion of long parts. There was some substitution of short part production with increased waste as the combined model searched for longer part solutions.

Higher updating frequency in the combined model contributes to a better yield although some exceptions were observed, especially in cutting list 2 as shown in Table 3. It is also interesting to note from Table 2 that lower yields by the combined model for all the three scenarios (N=25, N=15 and N=5) were measured for cutting list 5 when compared with the yield obtained by the goal-seeking model. This is likely due to a "trade-off" effect for the model to maintain a low level of overproduction.

Table 5 reveals that lower overall costs can be achieved by the combined model with high updating frequency (N=5) when compared to corresponding results obtained by the goal-seeking model. The cost savings by the combined model are more significant when hard-to-get cutting lists (in this case, the cutting lists 4 and 5) are processed. This benefit is contributed mostly by the model's effective control to overproduction and it becomes more significant when the inventory penalty is high. However, the results obtained are clearly sensitive to the overproduction penalty assigned to the model.

The solution updating frequency is crucial to the performance of the combined model. High updating frequency will achieve a lower amount of overproduction, a higher part yield and ultimately a lower overall cost.

The combined-model developed in this study shows that by using the adaptive methods the knowledge of the grade of rough lumber and the knowledge of the inventory levels for each part produced in a production run can be collected in a real time process. The model can meet the part order requirements using less raw material, producing less unneeded parts, and minimizing the overall cost for the production run.

CHAPTER 5

5.1 SUMMARY

Two chop saw optimization models using adaptive optimization methods were successfully created. Overall cost was set up as an indicator to evaluate the performance of the models. Two simulation studies were conducted to test the performance of the models. The major limitation of these models is that they can only process parts with same thickness and width.

The goal-seeking model and the combined model developed in this project show that by using adaptive methodology the knowledge of the quality of the incoming boards and the inventory level for each part can be obtained along with the production progress. This allows the models able to adapt to the changes on the plant floor and therefore they could be implemented in a real-time control process.

The simulation study indicates that by improving the optimization algorithm to solve the chop saw optimization problem, higher part yield, lower overproduction level and ultimately less overall cost can be achieved. In addition, all the parts on the cutting list are produced in a balanced rate along with the production process. By the time the production run is done, all the parts are chopped on time with limited amount of overproduction.

The simulation study for the goal-seeking model found that no single prioritization strategy can respond effectively to all the cutting lists. Of the two prioritization strategies developed in this study, SDV performs well for the "easy-to-get" cutting lists while CDV performed well for hard-to-get cutting lists.

Updating frequency is critical to the performance of the combined model in terms of yield, overproduction and overall cost. Higher updating frequency yields a better result than lower updating frequency.

In contrast to the goal-seeking model with two different prioritization strategies, the combined model has good performance for all the five cutting lists used in this simulation study.

5.2 RECOMMENDATIONS

The study used a specific data set throughout. Future research should look at the performance of the two models in a real-time situation. Moreover, the performance of the models should be tested against multiple grades of rough lumber.

Room to improve the performance of the goal-seeking model exists. It was noticed that the last pieces remaining on the "hard-to-get" cutting list in the near end of the production run were always long parts. It is obvious that if, at this moment, the incoming blank is shorter than any of the rest parts, the only option left for the model is to overproduce short parts. Therefore, future research should be geared towards the elimination of this phenomenon. It is recommended that more prioritizing power should be given to long parts on those hard-to-get cutting list so as to produce them in the production run. This objective could be achieved by incorporating different prioritization strategies into one optimization program. This program could adapt to grade of boards used for the production run and different cutting lists by automatically switching from one strategy to another.

Further investigation of the combined model is needed to test how frequent the LP updating (N) could be for a given cutting list in order to get the best result in real time.

The combined model could be also modified to solve the panel cutting problem in the furniture industry. As this is a typical Two Dimensional Cutting Stock problem, the DP algorithm in the combined model should be extended to consider the raw panel and parts with two dimensions (length and width) instead of just one dimension (length) as it does in the original model. The adaptive methods would no longer be applicable in this case as the dimension of the raw panels (For example, 4'×8') to be cut is known before the production run starts. The modified model could either be used in an on-line control system or be used as a production planning tool to optimize the cutting decision over multiple periods.

Based on either the goal-seeking model or the combined model, an integrated rip-first optimization model could be developed which can optimize the cutting of rough lumber for a rip-chop process.

6.0 REFERENCES

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

Buehlmann, U. 1998. Understanding the relationship of lumber yield and cutting requirements: a statistical approach. http://scholar.lib.vt.edu/theses/available/etd91298-173331/unrestricted/diss bue.PDF. Unpublished PhD thesis. Virginia Polytechnic Institute and State University, Blackburg, VA, USA. 207 P.

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

Chow, G.R. 1998. A comparison of twenty simulated strategies for achieving maximum parts value recovery in optimizing chop saw operations. Master's thesis, Univ. of British Columbia, Vancouver, BC, Canada. 106 P.

Dantzig, G. B. & Wolfe, P. 1960. Decomposition principle for linear programming. Operations Research. 8:101-111.

Dmitrovic, B. and D. Wolf 1992. A standard model of the placement problem aimed at resolving a cutting stock problem. IEEE Transactions in industry application. 28(6):1406-1415

Dreyfus, S. E. & Law, A. M. 1977. The art and theory of dynamic programming. Academic Press, Inc. ISBN 0-12-221860-4. 284 P.

Foronda, S.U., and H.F. Carino. 1991. A heuristic approach to the lumber allocation and manufacturing in hardwood dimension and furniture manufacturing. European J. Operations Res. 54:151-162

Gatchell, C.J. 1987. Rethinking the design of the furniture rough mill. Friest Prod. J. 37(3):8-14

Ghodsi R., J.A. Meech, D.B. Kotak. and F. Sassani. 2001. A rule-based expert system for cut sequencing of solid wood for furniture components production. Proceedings of the Third International Conference on Intelligent Processing and Manufacturing of Materials (IPMM), July 29 to August 3, 2001, Vancouver.

Giese, P.J. and J.D. Dannielson. 1983. CROMAX - a crosscut-first computer simulation program to determine cutting yield. Tech. Rept. FPL-38. USDA Forest Serv., Forest Lab., Madison, WI.

Giese, P.J. and K.A. McDonald. 1982. OPTYLD-A multiple rip-first computer program to maximize cutting yields. Res. Pap. FPL-412. USDA Forest Serv., Forest Prod. Lab. Madison, WI.

Gilmore, P.C. and R.E. Gomory. 1965. Multi-stage cutting stock problems of twodimensions. Operations Research. 13:94-120

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

Hillier, F. S. & Lieberman, G. J. 2001. Introduction to operations research. McGraw-Hill Higher Education. ISBN 0-07-246121-7. 1214 P.

Maness, T. C. & Adams, D. M. 1991. The Combined optimization of log bucking and Sawing strategies. Wood and Fibre Science. 23(2), 296-314.

Maness, T.C. and D. Wong. 2002. A benchmarking system for evaluating the profitability and efficiency of optimizing chop saw systems. Forest Prod. J. 52(10):5261

Mendoza, G.A. and B. Bare. 1986. A two-stage decision model for log bucking and allocation. Forest Prod. J. 36(10):70-74

Mullin, S. 1990. Why switch to rip-first?. Furniture Design & Manufacturing Magazine 62(9): 36-42.

Sessions, J., R. Layton, and L. Guangda. 1988. Improving tree bucking decision: A network approach. The Compiler. 6(1):5-9.

Stern, A.B. and K.A. McDonald. 1978. Computer optimization of cutting yield from multiple ripped boards. Res. Pap. FPL-318. USDA Forest Serv., Forest Prod. Lab. Madison, WI.

Thomas, R.E. 1996. Prioritizing parts from cutting bills when gang-ripping first. Forest Prod. J. 46(10):61-66

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

Wengert, E.M. and F.M.Lamb. 1994. A hand book for improving quality and efficiency in rough mill operations: practical guidelines, example and ideas. R. C. Byrd Hardwood Technology Center, Princeton, WV.

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

APPENDIX I

RESULT SHEET OF GOAL-SEEKING MODEL –CUTTING LIST 1

Cutting List 1		*Yield (%)*						*Volume (m3)*						*Overall Cost ($)*					*Avg*
Prioritization Strategy		Order File					Avg	Overproduction					Avg						
Goal-Seeking Model	Simple Dynamic Value (SDV)	86.21	86.21	85.6	86.25	86.27	**86.11**	0.023	0.023	0.061	0.005	0.014	**0.025**	8051	8051	8118	8045	8043	**8062**
	Complex Dynamic Value (CDV)	85.23	85.22	85.07	85.08	85.34	**85.19**	0.004	0.004	0.003	0.003	0.009	**0.005**	8153	8153	8168	8167	8140	**8156**
Thomas's Model	Simple Dynamic Exponent (SDE)	84.83	85.30	85.58	85.68	85.61	**85.4**	0.111	0.059	0.009	0.025	0.017	**0.044**	8208	8151	8115	8107	8113	**8139**
	Complex Dynamic Exponent (CDE)	85.04	85.11	85.23	85.26	85.51	**85.23**	0.041	0.022	0.072	0.011	0.044	**0.038**	8175	8166	8159	8149	8127	**8155**

APPENDIX II

RESULT SHEET OF GOAL-SEEKING MODEL –CUTTING LIST 2

Cutting List 2		Yield (%)						Volume (m3)						Overall Cost ($)					Avg
Prioritization Strategy		Order File					Avg	Overproduction					Avg						
Goal-Seeking Model	Simple Dynamic Value (SDV)	86.03	85.59	85.8	86.18	86.11	**85.94**	0.044	0.056	0.039	0.008	0.038	**0.037**	8072	8119	8094	8053	8063	**8080**
	Complex Dynamic Value (CDV)	85.19	85.05	84.93	85.21	85.05	**85.09**	0.013	0.01	0.009	0.006	0.003	**0.008**	8156	8172	8184	8154	8171	**8167**
Thomas's Model	Simple Dynamic Exponent (SDE)	85.21	85.02	85.25	85.35	85.58	**85.28**	0.072	0.029	0.04	0.015	0.018	**0.035**	8163	8177	8154	8140	8116	**8150**
	Complex Dynamic Exponent (CDE)	84.89	85.25	84.91	85.18	85.27	**85.1**	0.074	0.015	0.044	0.04	0.018	**0.038**	8197	8150	8191	8162	8149	**8170**

APPENDIX III

RESULT SHEET OF GOAL-SEEKING MODEL –CUTTING LIST 3

Cutting List 3		Yield (%)						Volume (m3)						Overall Cost ($)					Avg
Prioritization Strategy		Order File					Avg	Overproduction					Avg						
Goal-Seeking Model	Simple Dynamic Value (SDV)	85.41	84.97	85.04	84.59	84.47	**84.9**	0.060	0.113	0.065	0.178	0.187	**0.121**	8137	8188	8176	8235	8250	**8197**
	Complex Dynamic Value (CDV)	85.19	85.07	85.29	84.92	85.12	**85.12**	0.000	0.000	0.006	0.013	0.003	**0.004**	8155	8167	8145	8185	8163	**8163**
Thomas's Model	Simple Dynamic Exponent (SDE)	85.04	85.26	85.27	85.33	85.16	**85.21**	0.081	0.053	0.008	0.022	0.036	**0.04**	8181	8154	8147	8143	8162	**8157**
	Complex Dynamic Exponent (CDE)	85.23	84.87	85.19	85.17	84.59	**85.01**	0.005	0.017	0.023	0.039	0.100	**0.037**	8151	8190	8157	8162	8230	**8178**

APPENDIX IV

RESULT SHEET OF GOAL-SEEKING MODEL –CUTTING LIST 4

Cutting List 4		Yield (%)						Volume (m3)						Overall Cost ($)					Avg
Prioritization Strategy		Order File					Avg	Overproduction					Avg						
Goal-Seeking Model	Simple Dynamic Value (SDV)	82.86	82.74	83.12	82.36	82.23	**82.66**	0.462	0.484	0.407	0.537	0.523	**0.483**	8450	8465	8416	8513	8527	**8474**
	Complex Dynamic Value (CDV)	84.89	85.44	85.28	84.95	85.26	**85.16**	0.010	0.015	0.015	0.015	0.007	**0.012**	8188	8131	8147	8183	8149	**8160**
Thomas's Model	Simple Dynamic Exponent (SDE)	84.77	84.96	84.94	85.08	85.08	**84.97**	0.135	0.056	0.065	0.023	0.011	**0.058**	8216	8186	8189	8169	8168	**8186**
	Complex Dynamic Exponent (CDE)	84.96	85.24	84.98	85.06	85.33	**85.11**	0.043	0.039	0.045	0.013	0.042	**0.036**	8184	8155	8183	8170	8146	**8168**

APPENDIX V

RESULT SHEET OF GOAL-SEEKING MODEL –CUTTING LIST 5

Cutting List 5		Yield (%)						Volume (m3)						Overall Cost ($)					Avg
Prioritization Strategy		Order File					Avg	Overproduction					Avg						
Goal-Seeking Model	Simple Dynamic Value (SDV)	79.29	79.79	79.57	79.37	80.29	**79.66**	0.996	0.942	0.952	1.000	0.875	**0.953**	8913	8848	8876	8905	8781	**8865**
	Complex Dynamic Value (CDV)	84.21	84.23	84.09	84.54	83.68	**84.15**	0.093	0.104	0.148	0.072	0.171	**0.118**	8297	8296	8305	8260	8332	**8298**
Thomas's Model	Simple Dynamic Exponent (SDE)	83.27	83.81	83.59	82.74	83.59	**83.4**	0.286	0.224	0.289	0.422	0.289	**0.302**	8396	8331	8362	8747	8362	**8440**
	Complex Dynamic Exponent (CDE)	83.34	82.91	83.34	83.14	83.46	**83.24**	0.232	0.367	0.347	0.278	0.311	**0.307**	8382	8448	8396	8408	8378	**8402**

APPENDIX VI

RESULT SHEET OF COMBINED MODEL –CUTTING LIST 1 TO 5

Cutting List 1		*Yield (%)*			*Volume (m3)*			*Overall Cost ($)*		*Avg*
		Order File		Avg	Overproduction		Avg			
Combined Model	N=5	86.00%	86.29%	**86.15%**	0.021	0.016	**0.0185**	8072	8059	**8066**
	N=15	85.48%	85.35%	**85.42%**	0.084	0.0754	**0.0797**	8145	8127	**8136**
	N=25	85.28%	85.31%	**85.30%**	0.1058	0.1188	**0.1123**	8158	8165	**8162**

Cutting List 2		*Yield (%)*			*Volume (m3)*			*Overall Cost ($)*		*Avg*
		Order File		Avg	Overproduction		Avg			
Combined Model	N=5	86.28%	85.94%	**86.11%**	0.027	0.031	**0.029**	8071	8084	**8078**
	N=15	85.21%	85.43%	**85.32%**	0.06543	0.08712	**0.07628**	8183	8175	**8179**
	N=25	85.28%	85.12%	**85.20%**	0.1269	0.1345	**0.1307**	8167	8205	**8186**

Cutting List 3		*Yield (%)*			*Volume (m3)*			*Overall Cost ($)*		*Avg*
		Order File		Avg	Overproduction		Avg			
Combined Model	N=5	85.02%	85.22%	**85.12%**	0.052	0.049	**0.0505**	8182	8158	**8170**
	N=15	84.17%	84.80%	**84.49%**	0.0872	0.07241	**0.07981**	8260	8241	**8251**
	N=25	84.12%	84.84%	**84.48%**	0.0923	0.1388	**0.11555**	8279	8234	**8257**

Cutting List 4		*Yield (%)*			*Volume (m3)*			*Overall Cost ($)*		*Avg*
		Order File		Avg	Overproduction		Avg			
Combined Model	N=5	85.23%	85.02%	**85.13%**	0.031	0.052	**0.0415**	8155	8205	**8180**
	N=15	84.54%	84.43%	**84.49%**	0.0715	0.0915	**0.0815**	8234	8245	**8240**
	N=25	84.14%	84.09%	**84.12%**	0.1175	0.1243	**0.1209**	8283	8297	**8290**

Cutting List 5		*Yield (%)*			*Volume (m3)*			*Overall Cost ($)*		*Avg*
		Order File		Avg	Overproduction		Avg			
Combined Model	N=5	83.37%	83.49%	**83.43%**	0.048	0.039	**0.0435**	8334	8305	**8320**
	N=15	82.94%	83.01%	**82.98%**	0.1072	0.01009	**0.05865**	8409	8394	**8402**
	N=25	82.87%	82.46%	**82.67%**	0.1514	0.01678	**0.08409**	8421	8453	**8437**