Dissertationes Forestales 37

Design and testing of stand-specific bucking instructions for use on modern cut-to-length harvesters

Veli-Pekka Kivinen
Department of Forest Resource Management
Faculty of Agriculture and Forestry
University of Helsinki

Academic dissertation

To be presented with the permission of the Faculty of Agriculture and Forestry of the University of Helsinki for public examination in Lecture Room 3, B-building, Latokartanonkaari 9, Helsinki, on April 27th 2007, at 12 noon.

Title of the dissertation:
Design and testing of stand-specific bucking instructions for use on modern cut-to-length harvesters

Author:
Veli-Pekka Kivinen

Dissertationes Forestales 37

Supervisors:
Docent Jori Uusitalo,
Finnish Forest Research Institute, Parkano Research Unit, Finland

Professor Esko Mikkonen
Department of Forest Resource Management, Faculty of Agriculture and Forestry, University of Helsinki, Finland

Pre-examiners:
Professor Maarten Nieuwenhuis,
School of Biology & Environmental Science, University College Dublin, Ireland

Professor Reino E. Pulkki
Faculty of Forestry and the Forest Environment, Lakehead University, Canada

Opponent:
Professor Teijo Palander
Faculty of Forestry, University of Joensuu, Finland.

ISSN 1795-7389
ISBN 978-951-651-162-0 (PDF)

(2007)

Publishers:
The Finnish Society of Forest Science
Finnish Forest Research Institute
Faculty of Agriculture and Forestry of the University of Helsinki
Faculty of Forestry of the University of Joensuu

Editorial Office:
The Finnish Society of Forest Science
Unioninkatu 40A, 00170 Helsinki, Finland
http://www.metla.fi/dissertationes

Kivinen, V.-P. 2007. Design and testing of stand-specific bucking instructions for use on modern cut-to-length harvesters. University of Helsinki, Department of Forest Resource Management.

ABSTRACT

This study addresses three important issues in tree bucking optimization in the context of cut-to-length harvesting. (1) Would the fit between the log demand and log output distributions be better if the price and/or demand matrices controlling the bucking decisions on modern cut-to-length harvesters were adjusted to the unique conditions of each individual stand? (2) In what ways can we generate stand and product specific price and demand matrices? (3) What alternatives do we have to measure the fit between the log demand and log output distributions, and what would be an ideal goodness-of-fit measure?

Three iterative search systems were developed for seeking stand-specific price and demand matrix sets: (1) A fuzzy logic control system for calibrating the price matrix of one log product for one stand at a time (the stand-level one-product approach); (2) a genetic algorithm system for adjusting the price matrices of one log product in parallel for several stands (the forest-level one-product approach); and (3) a genetic algorithm system for dividing the overall demand matrix of each of the several log products into stand-specific sub-demands simultaneously for several stands and products (the forest-level multi-product approach).

The stem material used for testing the performance of the stand-specific price and demand matrices against that of the reference matrices was comprised of 9 155 Norway spruce (*Picea abies* (L.) Karst.) sawlog stems gathered by harvesters from 15 mature spruce-dominated stands in southern Finland. The reference price and demand matrices were either direct copies or slightly modified versions of those used by two Finnish sawmilling companies. Two types of stand-specific bucking matrices were compiled for each log product. One was from the harvester-collected stem profiles and the other was from the pre-harvest inventory data.

Four goodness-of-fit measures were analyzed for their appropriateness in determining the similarity between the log demand and log output distributions: (1) the apportionment degree (index), (2) the χ^2 statistic, (3) Laspeyres' quantity index, and (4) the price-weighted apportionment degree.

The study confirmed that any improvement in the fit between the log demand and log output distributions can only be realized at the expense of log volumes produced. Stand-level pre-control of price matrices was found to be advantageous, provided the control is done with perfect stem data. Forest-level pre-control of price matrices resulted in no improvement in the cumulative apportionment degree. Cutting stands under the control of stand-specific demand matrices yielded a better total fit between the demand and output matrices at the forest level than was obtained by cutting each stand with non-stand-specific reference matrices. The theoretical and experimental analyses suggest that none of the three alternative goodness-of-fit measures clearly outperforms the traditional apportionment degree measure.

Keywords: harvesting, tree bucking optimization, simulation, fuzzy control, genetic algorithms, goodness-of-fit

PREFACE

When I was starting my forestry studies in the mid 1980s, I never planned to become a researcher. Neither did I ever plan to write a Ph.D. thesis. My sincere wish was to graduate quickly and get a good job in some wood procurement company in Finland. My intentions were not realized, however. I do have done research work for the last thirteen years. I did write this Ph.D. thesis. What happened?

There was a deep economic depression in Finland in the early 1990s. At that time, I was still an undergraduate, wondering what to do with my life. At the same time, Jori Uusitalo was doing his Ph.D thesis at the Department of Forest Resource Management, University of Helsinki. Esko Mikkonen, a professor of forest technology and also the supervisor of Jori’s work, knew my situation and recommended Jori to hire me for his research team. I was fortunate enough to get a research assistant position in Jori’s thesis project, and the rest is history.

I thank you Jori for all the guidance, support and patience you have given me during the thirteen years we have been working together. You have not only been my supervisor but also a teacher, business partner, and friend. I guess if I had not met you, I would not be here as a researcher and Ph.D. candidate.

Thank you Esko for introducing me to Jori, for assisting me in many practical issues, and for always having time to listen to my academic and non-academic worries.

The Department of Forest Resource Management has been a large part of my life for the last 20 years, first as a student and then as a researcher. I thank Marketta Sipi, the head of the department, and Rihko Haarlaa, the former head of the department, for providing an ideal environment for my research and writing. Many thanks to Raili Onnela and Katriina Toivonen for administrative services and to Martin Ericsson and Johan Holmström for providing excellent IT support. Special thanks go to Hannu Rita for assisting me with the additional analyses included in the summary part. The entire personnel of the department is thanked for the friendly and helpful atmosphere and many inspiring discussions over lunch and evening coffee breaks over the years.

This study was carried out as a part of the three collaborative research projects. The first, starting in April 1998, was a sub-project of a WoodWisdom research programme consortium including the University of Helsinki (UH), the University of Joensuu (JOY), Helsinki University of Technology (HUT), and the VTT Technical Research Centre of Finland. The two other projects were jointly carried out by the universities of Helsinki, Joensuu and Tampere (UTA). I thank the following people for their pleasant and productive co-operation: Tapio Nummi, Laura Koskela, Anne Puustelli, Jarkko Isotalo, and Erkki Liski from the Department of Mathematics, Statistics and Philosophy (UTA); Tuomo Nurminen from the Faculty of Forestry (JOY)/Forest Agency Tuomo Nurminen; Heikki Korpunen from the Finnish Forest Research Institute (Metla); and Arto Usenius and Jorma Fröblom from VTT.

Many other people have contributed to this study. My pre-examiners, Maarten Nieuwenhuis and Reino E. Pulkki, provided constructive and thoughtful comments and suggestions on the manuscript of the summary part. Their feedback clearly made the summary part much stronger and certainly much more readable. Harri Kalola from Koskitukki Oy and Teppo Oijala and Toivo Vehmaanperä from Metsäliitto assisted me in many ways during the data collection phases. The anonymous harvester operators working in the study stands kindly saw to the collection of stem data files. Jari Korhonen from Ponsse Oyj was always willing to answer my (silly) tricky questions. Roderick McConchie

from the English Department (UH) carefully revised the language of both the summary part and the four articles. Arto Kettunen, my friend, fellow student and colleague, was the first who introduced me to genetic algorithms. I thank you all for your help.

I am grateful for research funding from the Ministry of Agriculture and Forestry, the Academy of Finland, the Niemi Foundation, and the Finnish Cultural Foundation.

I dedicate this study to my mother, father, and brother. There have been many heavy moments in your lives over the years. I do hope my work will serve as a source of joy, happiness and strength for you. Thanks for your constant love, support, and understanding.

Loppi, February 2007

Veli-Pekka Kivinen

LIST OF ORIGINAL ARTICLES

This thesis consists of this summary and the following four articles, referred to in the text by the Roman numerals I-IV:

I Kivinen, V.-P. & Uusitalo, J. 2002. Applying fuzzy logic to tree bucking control. Forest Science 48(4): 673-684.

II Kivinen, V.-P. 2004. A genetic algorithm approach to tree bucking optimization. Forest Science 50(5): 696-710.

III Kivinen, V.-P. 2006. A forest-level genetic algorithm based control system for generating stand-specific log demand distributions. Canadian Journal of Forest Research 36(7): 1705-1722. doi: 10.1139/X06-055.

IV Kivinen, V.-P., Uusitalo, J. & Nummi, T. 2005. Comparison of four measures designed for assessing the fit between the demand and output distributions of logs. Canadian Journal of Forest Research 35(3): 693-702. doi: 10.1139/X04-196.
Erratum: Eq. 3 on page 695 is incorrect. The correct form of Eq. 3 is:

$$\chi^2 = \sum_{i=1}^{m}\sum_{j=1}^{n}\frac{\left(o_{ij} - e_{ij}\right)^2}{e_{ij}} .$$

Study I: The study idea was conceived by Dr. Uusitalo, who also provided the basic guidelines for implementing the study. Kivinen did all the data acquisition work, planned and programmed the fuzzy logic control system, did the bucking simulations and analyzed the results. The original manuscript was written together, while its revised versions are mainly by Kivinen.

Study IV: The study was planned together by all three authors, who all contributed to analyzing the requirements for an ideal goodness-of-fit measure and the advantages and disadvantages of the four fitness measures introduced and tested in the study. Kivinen conducted all the experimental tests and analyzed the results. The original manuscript and its revised versions were written mainly by Kivinen.

Articles I and II are reproduced with the permission of the Society of American Foresters (SAF). Articles III and IV are reproduced with the permission of the National Research Council of Canada (NRC).

TABLE OF CONTENTS

1 INTRODUCTION

1.1 Bucking optimization

1.1.1 General

In order to be suitable for further processing, felled trees usually need to be converted into shorter logs. This operation is commonly called “tree bucking”, “log making”, or “log merchandising” (Marshall 2005) and results in various round wood products, such as sawlogs, veneer logs, poles, pulpwood logs, etc. Depending on the harvesting method and the subsequent delivery system employed, bucking can be done either directly in the stump area (i.e., on site), at the roadside, at a separate landing, at a centralized wood processing yard, or in a mill yard, or can be left completely undone as is the usual case in the chipping harvesting systems (Owende 2004, Pulkki 1997). In Scandinavia, where the cut-to-length (CTL) system is clearly the dominant harvesting method, the trees are almost always processed into the final log products at the stump while in North America, for example, the roadside and mill processing of full trees and tree-lengths is still widely used (Godin 2001, Greene et al. 2001, Owende 2004, Marshall 2005). Most of the bucking work, at least in the industrialized countries, is nowadays carried out mechanically by various types of processors, harvesters and stationary cutting equipment. Although manual bucking is generally restricted to harvesting work by forest owners, the motor-manual systems (i.e., bucking with a chainsaw) are still used in industrial wood procurement. For example, at some sites, the trees may simply be too large for typical mechanical processing (MacDonald 1999). Motor-manual processing is also generally preferred in harvesting tree species sensitive to mechanical damage (e.g., cutting birch logs for veneer).

Whether the bucking process takes place in the forest or at a mill yard, and whether it is done manually or mechanically, the key question remains the same: what log types (i.e., timber assortments), lengths, diameters, grades (qualities) and other attributes should a tree stem be cut into?

The answer to this question can be regarded as one of the most important decisions in timber harvesting and in the whole wood supply chain from forest to final customer/consumer product. This is simply because the bucking outcome in most conversion modes has a crucial effect on the profitability of the whole business (Usenius 1986). This arises from two well-known facts: (1) the properties of the resulting logs to a large extent determine what end products and quantities can be produced from a stem and thus the value of the stem (see, e.g., Fobes 1960, Smith and Harrell 1961); and (2) a poor bucking outcome is difficult or even impossible to compensate for at the subsequent manufacturing stages. This is especially true in mechanical wood processing, particularly sawntimber production, where there is a direct connection between the wood raw material and the end products. Thus, all knots and defects (e.g., rot, blue stain, bark and resin pockets, various shakes, etc.) that are present in a sawlog are also likely to be present in the lumber sawn from the log This often results in significant value losses through reduction in either the lumber volume (because of trimming losses, for example) or the lumber value (because of downgrading), or both. On the other hand, as logs of various types, sizes and grades are usually paid different amounts on the market, a forest owner, whether he/she is selling timber as standing or delivery sales, will lose money if bucking is done incorrectly or poorly.

As well as being important, determining an optimal bucking pattern (i.e., an optimal sequence of bucking cuts) for a tree stem is also one of the most challenging operations in timber harvesting for several reasons.

First, as is well known, trees are not regular in shape or homogeneous in their internal structure from the butt to the top. The main problem, however, is that for many reasons the geometry and the internal properties of tree stems are often poorly known or even totally unknown at the time of bucking.

For productivity reasons, it is usually uneconomic to run the whole stem through the processing/measuring device twice: first to measure the whole stem from the butt to the top and then buck it into the lengths determined according to this measurement data. Marshall (2005), for example, reports that the full scanning of the tree profile reduced the overall productivity of a mechanized forest harvester by a quarter to a third from that of the conventional harvesting system in which the measuring take places simultaneously with the delimbing and cutting processes. While there are already plenty of scanning technologies available for capturing internal features of tree stems such as X-ray technologies, ultrasonic measurement and nuclear magnetic resonance imaging (Nordmark and Oja 2004; Schmoldt et al. 2000), there may not be appropriate software tools available to process and analyze the huge amount of data typically gathered in the scanning process automatically, accurately and sufficiently quickly (in real time) (Schmoldt et al. 2000). Another problem is that most of the log/stem scanners were originally developed for operation in a mill environment and are thus either too large in size or too sophisticated to be used in harsh forest environments. As is the case with all measurements in general and tree measurements in particular, the measurement data does not usually come without errors. Thus, even if a tree stem is fully scanned from the butt to the top, the bucking pattern chosen may still be sub-optimal because of errors in stem information. Because the bucking decisions are frequently made with incomplete and erroneous information, it is no surprise that large value recovery losses have been reported worldwide to occur in both manual and mechanized log-making (Murphy and Olsen 1988, Garland et al. 1989, Olsen et al. 1991, Bowers 1998, Murphy 2002, Boston and Murphy 2003).

Second, as Kärkkäinen (1986) and Sessions (1988) state, the definition of what constitutes an optimal bucking pattern depends on the viewpoint of the decision-maker. A forest industry company, which buys the timber from forest landowners, harvests the timber and processes it into final end products, behaves like any company in any other sector; that is, it tries to maximize its profit. This means that each tree length should be cut into log lengths in such a way that the total net value of the end products produced from the stem is maximized. A forest landowner, on the other hand, usually wants to extract the maximum income from harvesting his/her forest resource. Because the sawlogs, veneer logs and other logs intended for use in mechanical wood processing are usually much higher in price than conventional pulpwood logs, the forest landowner thus seeks to minimize the amount of pulpwood from each stem. The problem is now that the bucking pattern maximizing a forest owner’s profit may not do the same for a timber buyer’s profit. This is especially the case when the timber pricing system is based on fixed, product-specific log prices (e.g., €/m^3), allowing neither premiums for high-quality logs or penalties for poor-quality logs.

Third, it is important to note that, from the viewpoint of a forest industry company, stem-level bucking optimization does not necessarily result in an optimal log output at the stand level, nor does stand-level optimization result in a forest-level optimum (Pickens et al. 1997, Laroze 1999, Arce et al. 2002). Surely, the situation would be different, if various end products and thus various roundwood products (i.e., logs of various types) were subject to no demand constraints derived from the market. However, different customers tend to have different needs concerning the amounts, types and characteristics of the products they are willing to buy. Because the product specifications largely determine the characteristics of logs to be supplied, the optimal bucking pattern for each tree in each stand should be actually determined by customer order(s) rather than the conventional goal of maximizing the value of each tree to be harvested. Further, several stands are usually required to meet each customer order. Because each stand usually represents a unique composition of trees in terms of number, size and quality, an overall (forest-level) optimal bucking policy can be achieved only by considering the production potential of each stand simultaneously with, rather than independently from, that of other stands.

Fourth, an average-size tree stem may easily have hundreds or even thousands of different feasible bucking patterns from which to select the optimum. Thus, even if we had complete knowledge of the external (and internal) characteristics of each tree in each stand to be felled, deriving an optimal bucking policy even for one stand would still be a computationally very demanding task. This is nicely demonstrated by Näsberg (1985, p. 34-37) who calculated the number of possible cutting patterns as a function of the merchantable timber length and the number of available log lengths. Assuming that the whole timber length from the butt to the minimum small-end diameter point (SED) is exploited as fully as possible and all the available log lengths, ranging from 34 to 55 dm at an interval of 3 dm, can be cut from any part of the merchantable stem section, the number of feasible bucking patterns is 12 348 for a tree length of 15 m and 499 202 for a tree length of 20 m. That is, if a stand comprises 500 identical trees each showing a 15 m long merchantable timber section, the search space of the optimal bucking policy for that stand consists of $12\,348^{500}$ ($\approx 2^{6800} \approx 10^{2048}$) different bucking alternatives. This being the case, it is quite obvious that the complete enumeration of all feasible bucking patterns, while it may work well for stem-level bucking optimization, is an absolutely inappropriate technique for solving stand- and forest-level bucking optimization problems efficiently. This is especially true if all the trees in each stand are treated as individuals rather than classified into a few categories defined, for example, by diameter or height or both.

1.1.2 Solution approaches at different levels

1.1.2.1 Stem level

The goal in stem-level bucking optimization is to assign each tree to be cut a bucking pattern yielding the highest total stem value. This requires that (1) the stem profile for the whole merchantable length from the butt to the minimum small-end diameter (SED) point be known and (2) each feasible length-diameter-quality combination of logs be given a value reflecting its profitability and/or desirability on the market. The individual log values can be either gross or net values derived from the sales income and production costs of various end products, or present log market prices on either an absolute or relative scale (see Näsberg 1985 p. 44-52). The first prerequisite enables the enumeration of all feasible bucking alternatives for the entire tree length, while the second makes it possible to assign an economic value (e.g., profit, value added, etc.) to each alternative generated. The principle of cutting a tree stem into logs with the highest aggregate value is commonly called bucking to value (Sondell 1987) or buck to value (Marshall 2005) while the actual problem of finding a bucking pattern with the maximum stem value is often referred to as a marking for bucking problem (MBP) (Näsberg 1985).

Several mathematical programming models to determine an optimal bucking pattern for a single tree stem have been introduced. Since Näsberg (1985) offers an excellent review of the various techniques and modeling approaches applied, the following brief review of these owes much to his work.

Most of the developed models for stem-level bucking optimization are clearly based on dynamic programming (DP). In general, dynamic programming is a solution approach to decision problems which are either inherently composed of or can be decomposed into sequential, interdependent stages, each with several alternative states (Anderson et al. 1994). This is exactly the case with the stem-level bucking-to-value optimization: the cut numbers (i.e., the log numbers given in increasing order from the stump) correspond to stages and the state space for each stage consists of all log lengths available for each log product involved in the optimization process. Dynamic programming is favored as a modeling approach mainly because of its computational efficiency. The DP formulation's better performance over the implicit enumeration technique is well illustrated by the following simple example from Laasasenaho (1996). Suppose that (1) we have a tree stem which is to be cut into four sawlogs (pulpwood logs excluded), (2) the available log lengths range from 37 to 64 dm at an interval of 3 dm, and (3) all possible length-diameter combinations of logs are permissible and thus have non-negative values (prices). The optimal bucking pattern in this particular case (see Fig. 1) can be found through the following 5-step DP procedure:

Step 1: Calculate the value of each of the 10 possible butt logs.
Step 2: Calculate the value of each of the 100 2-log combinations (10 butt log lengths x 10 2nd log lengths) and choose the best for each of the 19 potential cutting points: 74, 77, 80,…, 128 dm.
Step 3: Calculate the value of each of the 190 3-log combinations (19 possible starting heights x 10 possible log lengths for the 3rd log) and choose the best for each of the 28 potential cutting points: 111, 114, 117,…, 192 dm.
Step 4: Calculate the value of each of the 280 4-log combinations (28 possible starting heights x 10 possible log lengths for the 4th log) and choose the best for each of the 37 potential cutting points: 148, 151, 154,…, 256 dm (note that not all of these 37 potential cutting points may be feasible and therefore need not be considered in the calculations). The highest value of these 37 best values is the value of the optimal solution and the 4-log combination yielding this highest value represents the optimal bucking pattern.
Step 5: Determine the whole optimal bucking pattern (sequence of log lengths) by tracing back through the calculations made in the previous four steps. For example, the optimal length of the 3rd log is given by first subtracting the length of the 4th log from the total length of the optimal 4-log combination and then picking up the best 3-log combination for this remaining stem length from the results of step 3.

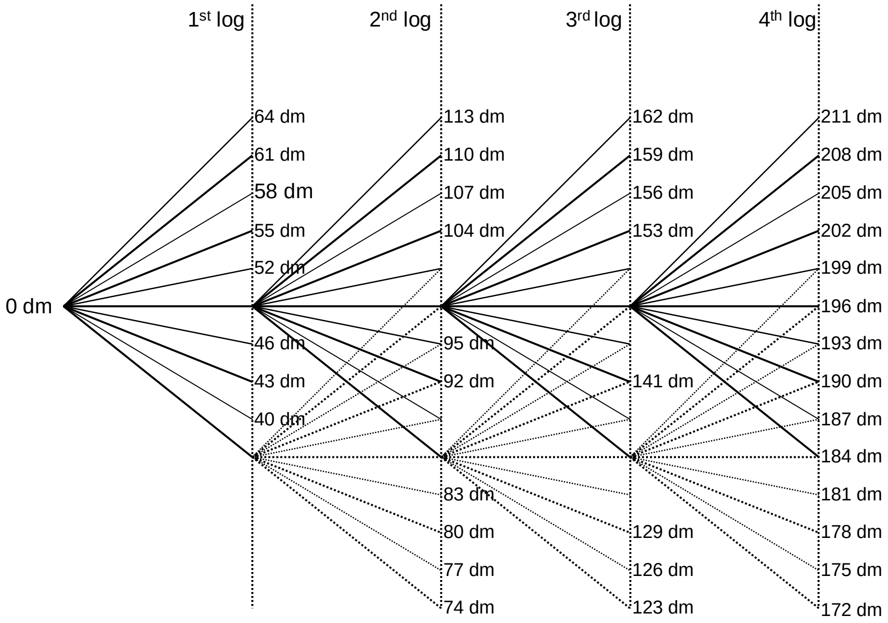

Figure 1. An optimal bucking pattern for a 4-log tree stem can be found efficiently by dividing the problem into four sequential sub-problems (finding an optimal log combination for each possible crosscutting point at each of the four log combination levels) and solving each using the optimal solutions of the previous sub-problem.

The theoretical number of possible bucking patterns for this 4-log tree stem is as high as 10 000 (= 10 x 10 x 10 x 10). In practice, the number is much smaller because the small-end diameter of the log combinations' last log is often likely to be below that of the minimum SED requirement. However, when employing total enumeration as a solution strategy for optimal bucking, we certainly would have to evaluate a large number of different bucking patterns to find the optimal one. If we solve the example using the DP approach, the number of evaluations needed for an optimal solution would drop dramatically from 10 000 to 570 (= 100 2-log combinations + 190 3-log combinations + 280 4-log combinations). This makes a complete enumeration technique under DP a practicable option.

The first detailed DP formulation for stem-level bucking optimization was introduced in the early 1970s by Pnevmaticos and Mann (1972). The idea of using DP as a solution approach had, however, already been introduced in the 1960s by Clemmons (1966) and Strand (1967), as cited by Näsberg (1985), Puumalainen (1998) and Wang et al. (2004). The main difference between these two DP models is the definition of the stages (sub-problems) the original master problem is divided into: in Strand's formulation the stages correspond to the cut numbers (i.e., log numbers) while Pnevmaticos and Mann divided the stem into segments of equal length, these segments then being associated with the stages.

Because the stem segment length in Pnevmaticos and Mann's model is equal to the minimum accepted log length, all log lengths are actually restricted to integer multiples of the shortest log. This restriction obviously requires that the minimum log length be unrealistically small (e.g., 3 dm), otherwise it may be impossible to include all available log lengths in the optimization process. Thus, in further developing the model of Pnevmaticos and Mann, Glück and Koch (1973) redefined the stages to correspond to the cut numbers (i.e., the approach Strand applied) while Briggs (1977, 1980) redefined the segment lengths as equal to the greatest common divisor of all available log lengths (usually 5 or 10 cm). Glück and Koch as well as Briggs also made other improvements to Pnevmaticos and Mann's model: (1) the log value was determined as a function of the log volume (or the volume of various end products produced from the log) rather than as a function of the log length only; (2) the quality of each log was determined in a deterministic rather than a stochastic way; and (3) the stem taper was, or at least could be, described using more realistic taper equations than the conventional truncated cone formula. Similar DP models for stem-level bucking optimization have been used by Faaland and Briggs (1984), Grondin (1998) and Reinders (1989) in developing integrated models for optimal tree utilization (i.e., models that integrate bucking optimization and log breakdown optimization).

These DP models, like DP models in general, are recursive and are thus often implemented through recursive algorithms. That is, a sub-problem at stage n is solved, using the optimal policy at stage n-1. Similarly, an optimal policy for sub-problem n-1 cannot be determined until an optimal policy for stage n-2 is found. In this way the search for an optimal bucking pattern proceeds sequentially from one stage to another until the butt end of the tree (backward recursion) or the minimum SED position at the top of the tree (forward recursion) is reached, in which case the search process terminates and an overall optimal solution can be constructed from the optimal solutions to the sub-problems. While elegant, compact and easy to design, recursive algorithms are often computationally burdensome because each function call at each stage places a complete copy of the function's 'information' (e.g., parameter values, variable values, return address etc.) in a computer's stack memory. This memory allocation is not released until the algorithm has reached the ultimate termination point (i.e., either the butt end or the minimum SED point). Thus, if the segment length is given a common value of 5 or 10 cm, computing an optimal bucking policy for a large population of tree stems may take a long time.

A more efficient network-based DP model for stem-level bucking optimization was introduced by Näsberg (1985) in investigating the possibility of using Operations Research (OR) techniques for controlling the log output distributions from forest harvesters to match the mills' demand distributions. The basic formulation in Näsberg's model is the same as in the recursive DP models, with the merchantable stem length being divided into short segments, each of length δ. However, because each node between two adjacent segments represents a potential cutting point, a network can be constructed by combining each node with another by an arc if the distance between the two cutting positions corresponds to a valid log length. Starting from the stump height (stage k = 0; k= 0,…,N), the solution procedure first generates and evaluates all 1-log combinations, ending at stem heights of $L_{min}…L_{max}$ (L_{min} is the minimum log length and L_{max} the maximum log length) (Fig. 2). The procedure then moves to the next stage (k = 1) and again forms and values all feasible 1-log combinations, ending at stem positions $L_{min}+k\delta…L_{max}+k\delta$. In this way the algorithm

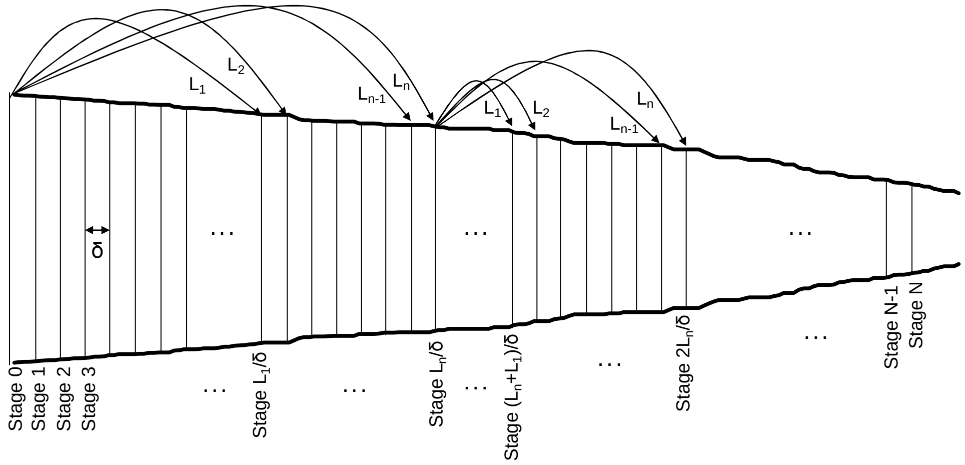

Figure 2. Network presentation for optimal log bucking. A tree stem is divided into N segments, each of length δ. At each stage k (k = 0…N), all valid log lengths $L_1,\ldots,L_n$ are tested, given the value (price) of each length-diameter(-quality) combination of logs. The optimal bucking pattern is found by recording the highest cumulative log value at each stage and the starting position of the last log in the best cutting pattern ending in stage k.

proceeds all the way towards the top of the tree until the stem diameter goes below the critical SED value. Näsberg (1985) termed this the longest route algorithm because its aim in essence is to find the most profitable path (the longest path) from the stump to the top of the tree. In practice this is done by means of two vectors: (1) one recording the highest cumulative log value for each potential cutting point (i.e., node) and (2) the other showing the starting position of the last log in each best cutting pattern ending at a particular node. Since Näsberg, similar kinds of network algorithms for stem-level bucking optimization have been proposed by many other researchers: e.g., Sessions et al. (1989), Wang et al. (1991, as cited by Wang et al. 2004), Puumalainen (1998), Sessions (1988), Gobakken (2000) and Wang et al. (2004).

Many decision-making, optimization and other types of problem can be modeled and successfully solved using linear programming (LP). The stem-level bucking optimization is no exception in this respect. Forster and Callahan’s model (1968, as cited by Bare et al. 1984 and Näsberg 1985) was presumably the first LP model for optimal stem conversion. Their objective is the maximization of the stem conversion surplus, given the conversion surplus (the market price of a log minus its procurement cost) associated with each feasible log-to-market alternative (i.e., each feasible log length-diameter-quality combination)). For this purpose, the tree stem is divided into 2-foot long segments, with the equal-to constraints requiring that each segment shall belong to some log-to-market alternative (i.e., the whole stem is to be exploited fully). As Bare et al. (1984) note, Forster and Callahan's LP model is (1) somewhat unrealistic because it assumes that all log lengths are multiples of 2 feet, (2) computationally inefficient because it requires the prior enumeration of all possible length-diameter-quality combinations (the number of various combinations can easily rise to several million (Bobrowski 1994)), and (3) somewhat ambiguous as regards to the incorporation of stem defects into the optimization process. Näsberg (1985) further points out that this model actually represents an integer linear programming (IP) model (0/1 integer linear model) rather than an LP model. This is because each 2-foot segment either belongs to a given log-to-market alternative (coded as 1) or not (coded as 0), with no in-between values being possible.

A different kind of IP model for stem-level bucking optimization was introduced by Näsberg (1985). The basis of his model is the concept of log classes: a log belongs to a log class (i,j) if the log's length *l* is greater than or equal to l_j but smaller than l_{j+1} (j = 1,…,n) and if the log's small-end diameter *d* is greater than or equal to d_i but smaller than d_{i+1} (i = 1,…,m). All logs with the same length and small-end diameter (SED) belong to the same log class (i,j) whatever their quality. However, logs with different qualities are associated with different prices, usually given in the form of price lists or matrices. The mathematical formulation of Näsberg's IP model is as follows:

$$\text{Max } \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} \qquad (1)$$

s.t.

$$\sum_{k=1}^{m}\sum_{j=1}^{n} l_j x_{kj} \le L_i \qquad \text{for all i} \qquad (2)$$

$$x_{ij} = \begin{cases} 1 & \text{if a log in log class (i, j) is cut from the stem} \\ 0 & \text{otherwise} \end{cases} \qquad \text{for all i,j}$$

m = number of small-end diameter (SED) classes
n = number of log lengths classes
L_i = distance from the stump to the position of stem diameter d_i
$c_{ij} = c_{ij}^{Q*}$
c_{ij}^{Q} = price of a log of quality Q in log class (i,j)
Q^* = quality of a log in log class (i,j) cut from the stem.

Beside the DP and LP models, an optimal bucking pattern for a single tree stem can be determined using the branch and bound method (BB). In fact, branch and bound, like dynamic programming, is not a specific solution technique but a solution approach applicable to a wide variety of problems (Taylor 1990). For example, the integer programming model above can be put into a branch and bound code which can then be solved in conjunction with the normal simplex method (Näsberg 1985). The first non-IP-based BB solution approach to stem-level bucking optimization was probably that of Ramalingham (1976) (see Näsberg 1985 and Bobrowski 1990, 1994). A similar kind of BB model was later suggested by Bobrowski (1990, 1994). Bobrowski (1994) also tested the performance of the BB approach and the conventional DP approach in terms of the CPU (central processing unit) time needed to arrive at an optimal solution (i.e., an optimal bucking pattern) and found the BB approach superior. In his test the solution time required by the BB model to derive an optimal bucking pattern for each of the 40 test trees in each of the 108 different test cases, for example, was always less than that of the DP model; the maximum CPU time ratio of DP to BB being as large as 20.

The main idea behind the branch and bound approach is the partition of the total solution space into smaller sub-spaces (sub-sets) of feasible solutions which are then evaluated systematically (Taylor 1990). When applied to the problem of converting a single tree stem into logs of various sizes and qualities in an optimal way, this partition principle results in a node-branch network (Fig. 3) similar to that of Fig. 1. Each node in the network represents a potential cutting point along the stem (the root node referring to the butt end of the tree) and has as many branches emanating from it as there are valid log lengths available. The process of generating new branches from each new node and attaching a new node to each new branch continues until a terminal node with branches not meeting the minimum requirements for the log length and small-end diameter is reached. Each set of branches (a path of branches) connecting the terminal node to the root node shows a feasible bucking pattern, with a total monetary return calculated from the individual log values included in the pattern.

Once the construction of the branch and bound node network for a tree stem is completed, a simple strategy for finding a bucking pattern yielding the highest total stem value would be to enumerate all potential bucking patterns along with their total stem values and select the pattern with the maximum value. Obviously, this is not the strategy employed by efficient branch-and-bound algorithms for optimal stem conversion. The efficiency of the BB algorithms is based on: (1) determining the lower and upper bound for the stem value at each node generated; and (2) pruning the infeasible and otherwise non-optimal solutions using these bounds (Bobrowski 1990, 1994). For example, given two nodes with the same remaining merchantable stem length for bucking (i.e., nodes located at the same height position from the butt end), the search for the optimal bucking pattern continues by branching from the node with the larger upper bound; i.e., the node with the smaller upper bound will be pruned out. Similarly, if the lower bound of one node exceeds the upper bound of the other, the previous node will be retained for further examination. The main problem with the BB-based bucking approach is that the potential value for the remaining stem length at each node must be estimated because considering all possible bucking patterns would simply take too much time (Bobrowski 1990). Poor value estimation can then result in premature pruning of the potentially optimal nodes, thus effectively obscuring the overall optimum.

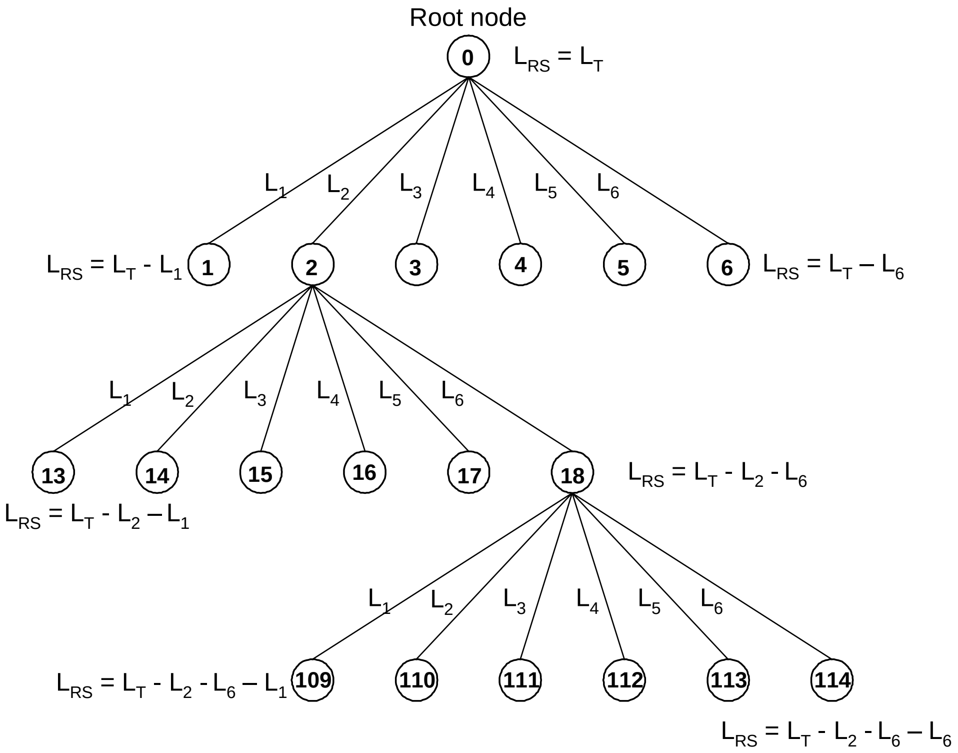

Figure 3. Root end of the branch and bound node network for optimally bucking a tree stem of merchantable length L_T into six alternative log lengths (L_1,…,L_6). L_{RS} stands for length of remaining stem (i.e., defining the distance between the current node and the stem position with the stem diameter equal to the minimum SED of logs).

The main assumption in the previously presented optimization approaches is that the stem profile (i.e., stem diameters from the butt end of the tree to the top) for the whole merchantable length of a tree stem is known, thus making it possible to determine the optimal bucking pattern for its whole length. Measuring the stem diameters at certain fixed intervals (e.g., at 1 m steps), however, may be too laborious. This is especially true in manual logging even though a logger may have a handheld data logger/computer to assist in data input and decision-making. Modern forest harvesters usually first feed and measure a tree stem for a short length (≤ a minimum feasible log length) and then predict the profile for the upper part of the stem. The problem may then be that the prediction model used is not capable of providing a sufficiently accurate profile estimate for the unknown stem section.

To address this problem, Imponen (1987) proposed that a near optimal bucking pattern can be easily derived using a step-by-step optimization procedure. Its main idea is that the optimization considers not the whole stem section from the butt to the smallest minimum SED but a shorter section consisting of two or three log lengths only. For each stem section of this length, all feasible bucking patterns along with their values are first created and the pattern with the highest aggregate value (i.e., the sum of the values of the logs included in the optimization) is selected for implementation. The whole optimal pattern is, however, not implemented, only the first log (i.e., the butt log) being cut as proposed by the pattern. Taking this first cutting point as a starting point, all feasible log combinations with their values are again listed for the next stem section composed of one or more log lengths, and the combination with the highest value is selected as optimal. The second log from the stem is then cut according to this second-stage optimal bucking pattern. The process continues in this stepwise manner until the entire merchantable stem is converted into short.

A stepwise bucking optimization algorithm, very similar to Imponen (1987), was also presented by Laroze and Greber (1997). Their model, as opposed to Imponen's approach, however, considers only one log at a time in the optimization calculation. The selection between various log candidates is made on the basis of (1) the characteristics of the stem being bucked, (2) the specifications for each log type, such as the minimum small-end diameter and the acceptable quality classes of tree stems, and (3) the priority list of log types. The priority list shows a complete enumeration of available log types arranged in descending order according to their net returns (a log with the highest profit is first, the lowest value log being the last). The algorithm, starting from the first log type (the highest value log type) in the priority list, evaluates whether the specifications of the proposed log are compatible with the characteristics of the current stem section. If not, the second log type in the priority list is analyzed. If a log of this type cannot be produced from the stem section being examined either, the third log in the priority list is then evaluated. This process continues until a log type that matches the characteristics of the current stem section is found. In this case, a log of the selected type is cut from the tree, after which the algorithm starts searching for the best bucking alternative for the next stem section. Although easy to implement, a greedy bucking algorithm of this kind may lead to serious sub-optimization because once a suitable log is found, it is bucked from the stem immediately without considering the effects of this decision on the subsequent bucking possibilities and thus the total net return from the stem.

1.1.2.2 Stand level

The goal in the stand-level bucking optimization is to find a bucking policy maximizing the aggregate production value from all stems being cut from a forest stand. As stated earlier, selecting a bucking pattern with the highest return for each tree stem in a stand may result in a severe mismatch between the desired log output distributions and the corresponding actual output distributions, and markedly lower overall profits. This is because logs not meeting the length, diameter and/or quality specifications of customer orders may need to be shortened or otherwise further processed to better match the end product requirements. If shortening of logs is not possible (because only large-sized logs can be converted into smaller ones), they are sold off on the open market. In both cases, some value losses are to be expected through the generation of extra waste, the extra cost caused by selling logs at prices possibly below the original purchase prices, and/or buying new logs at prices possibly higher than the original ones.

The process of determining an optimal bucking policy for a whole stand (i.e., a large set of individual tree stems) thus needs to consider not only the forest resource available, but also all the various merchandising restrictions imposed by the various end product markets and customers. Again, it should be noted that a bucking policy of this kind does not necessarily maximize a forest owner's harvesting income if the log prices employed in the optimization process are not real market prices and if the degree to which the actual log outcome satisfies the market demands has no effect on the final log purchase prices.

In general, bucking a large set of tree stems into smaller logs is analogous to many industrial cutting processes in that a large body of raw material is to be divided into smaller parts in an optimal way. A situation of this kind occurs, for example, in the paper industry where the trim width of a modern paper machine is around 9 to 10 m, while the width of printing machines typically varies between 1 and 3.5 m (Airola et al. 1999). This means that a paper roll from a paper machine usually needs to be slit and wound into several narrower rolls according to the unique width demand(s) of each customer. This obviously constitutes a decision problem: what would be the best cutting pattern for each large paper roll (i.e., a parent reel) to produce the required number of customer rolls. The simplest approach to this paper trim problem (PTP) intends to minimize the number of parent rolls needed to satisfy the customer orders. Assuming that only one parent roll width L is available, and allowing some overproduction of rolls while no withdrawals from any existing stock, this simple PTP approach can be formulated mathematically as follows (Eisemann 1957, Näsberg 1985):

$$\mathrm{Min}\sum_{j=1}^{n} x_j \tag{3}$$

s.t.

$$\sum_{j=1}^{n} a_{ij} x_j \geq N_i \quad \text{for all } i = 1,\ldots,m \tag{4}$$

$$x_j \geq 0 \text{ (and integer)} \quad j = 1,\ldots,n$$

$$L \geq w_i \qquad \text{for all } i = 1,\ldots,m$$

where

x_j = number of times cutting pattern j is used
a_{ij} = number of paper rolls of width w_i produced by cutting pattern j
N_i = demand for a paper roll of width w_i
L = width of the parent (large) roll
n = number of different cutting patterns (set-ups)
m = number of different paper roll widths (customer roll widths).

This problem, as Näsberg (1985) maintains, is hard to solve for two reasons. First, the decision variables x_j are assumed to take integer values only, because cutting patterns obviously cannot be implemented partially (i.e., each large roll must be cut into smaller rolls using one cutting pattern only). The restriction to integers, however, can be easily handled by simply dropping it; that is, the problem is treated as a continuous one and the final solution values are rounded either up or down to the nearest integers afterwards. This normally results in no serious sub-optimization if the activity levels in the LP model are large, as they usually are. The second problem is that in order to find an optimal set of cutting patterns for a given set of customer orders directly, all the thousands of feasible patterns (small roll combinations) should be explicitly included in the LP model. Although technically possible, the enumeration of all possible cutting patterns would certainly be a lengthy job. The main difficulty, however, lies in the fact that linear programming problems involving a large number of variables are intractable to solve, at least by means of the ordinary simplex method (Gilmore and Gomory 1961).

Large-size (integer) linear programming problems can, however, be solved without actually knowing all possible activities (columns) in the LP model in advance, as shown by Gilmore and Gomory (1961, 1963). Their solution approach to the cutting-stock problem (i.e., the problem of finding the least-cost cutting program to produce the desired numbers of pieces of lengths l_1, $l_2,\ldots,l_m$ from a stock of standard lengths L_1, $L_2,\ldots,L_n$ where $l_i \leq L_j$ for all i = 1,…,m and j = 1,…,n) was based on the implicit column generation method (Ford and Fulkerson 1958), and the Dantzig-Wolfe decomposition principle (Dantzig and Wolfe 1960, 1961). The main idea is to seek an optimal solution iteratively through a two-stage procedure rather than directly through the normal simplex computation. In practice, this is done by (1) generating a restricted number of feasible cutting patterns, (2) solving the integer-relaxed version of the original LP problem (a so-called restricted master problem) using this initial set of cutting patterns, and (3) checking for the optimality in solving the LP relaxation by solving an auxiliary problem (a so-called pricing problem), given the dual prices from the original (main) LP problem. In the case of this cutting-stock problem, the auxiliary problem is of the following form:

$$\text{Max } Z = \sum_{i=1}^{m} u_i y_i \tag{5}$$

s.t.

$$\sum_{i=1}^{m} w_i y_i \leq L \tag{6}$$

$y_i \geq 0$ (and integer) for all i = 1,…,m

where

u_i = dual cost of the customer roll width of w_i (from the original LP problem)
y_i = number of customer rolls of width w_i to include in the pattern.

The optimal solution to this problem, called a knapsack problem, identifies a new cutting pattern (i.e., a new combination of small paper rolls of widths w_i). This new pattern will be added to the restricted master LP problem as a new column if the optimal value Z > 1. The master LP problem along with its dual problem is then resolved, producing the new dual prices to be used in the knapsack problem to generate a new cutting alternative. Again, if the objective function value of the knapsack problem exceeds the value of 1, the new cutting alternative generated will be included as a new activity in the master problem. This iterative process continues until there are no new cutting patterns from the knapsack problem to include in the master problem. Because the knapsack problem is often solved using dynamic programming, this two-stage procedure is consequently referred to as a combined LP-DP method. If, on the other hand, a similar column generation technique is employed in conjunction with branch and bound (i.e., the original LP problem is being solved using branch and bound), the resulting solution approach is called branch and price (Hans 2001). More details on column generation and decomposition within the context of the paper trim problem (cutting-stock problem) can be found in Gilmore and Gomory (1961) and Näsberg (1985).

In tree bucking optimization, the first attempt to systematically calculate rather than intuitively judge the optimal set of bucking patterns (cut-up processes) for a single stand of timber was made by Smith and Harrell (1961). Their optimization approach was based on a standard linear programming (LP) technique, the activities in the LP model (i.e., decision variables) offering potential cutting patterns for different tree-size (DBH) classes. Because of the limited capacity of the computer resources available at the time of the study only three heuristically created cutting patterns were included in the optimization model for each tree class. Given the maximum number of trees available in each of the six classes, the minimum and/or maximum volume requirements for various log lengths, and the net profit from bucking a tree in a particular size class with a particular cutting pattern, the simplex method then iteratively searches for a bucking pattern combination that (1) satisfies all linear market and resource restrictions given, and (2) simultaneously maximizes the overall net profit from the harvesting operation (i.e., maximizes the difference between the total sales income from logs harvested and their logging and transportation costs).

Smith and Harrell’s LP-based optimization approach (1961), though it works smoothly technically, shares the same problems as the PTP model above (i.e., Equations 3 and 4). First, in order to find an absolutely optimal set of bucking patterns for a given stand, all feasible patterns for each tree-size class should be explicitly included in the LP model. However, as each tree may have hundreds or even thousands of different bucking patterns and as trees even in the same size class are seldom exact copies of one another, the number of different activities (bucking pattern – tree-size class combinations) and thus the size of the LP model may become enormous, especially if there are many short log lengths possible (Näsberg 1985). Second, because of representing the number of trees cut by a particular bucking pattern in a stem class, an activity in Smith and Harrell’s LP model obviously cannot take non-integer values.

The requirement that all possible bucking patterns for each stem-size class should be known in advance can be overcome by simply applying the indirect solution approach discussed above for optimally solving the integer PTP (cutting-stock) problem. This is exactly what was done by Eng and Daellenbach (1985), Eng et al. (1986), as well as Mendoza and Bare (1986) (see also Laroze and Greber 1997). All presented a price-directed two-stage optimization procedure in which the upper level of the model (i.e., the master problem) is formulated as an LP model, and the lower level model is formulated as a dynamic program. The optimization objective is to assign each stem-size class, defined, for example, by tree length and/or breast-height diameter, a bucking pattern or a set of bucking patterns to maximize either the overall market value of all logs produced (Eng and Daellenbach, Eng et al.) or the total net profit from wood end products produced from the logs cut (Mendoza and Bare). The procedure starts by finding – arbitrarily or using some heuristics – at least one feasible bucking pattern for each stem-size class. Using these initial bucking patterns, the master LP problem is then solved to determine the number of stems in each class to be bucked with each bucking pattern available (i.e., an optimal bucking policy for the whole stand). Given the shadow price or the Lagrange Multiplier of each log type from the upper level LP solution, the lower level DP problem is then solved for each stem class to see whether there may still be some new bucking patterns which could potentially improve the value of the objective function (i.e., the value of the optimal solution to the upper level LP problem). If there are, these are added to the upper level LP model as new columns (i.e., new activities), after which the LP problem is re-solved, resulting in new shadow prices or new Lagrange Multipliers to be used by the lower level DP procedure. If the DP procedure cannot recognize any new profitable bucking patterns for any stem class, the whole process stops, with the current LP solution being the optimum.

Pickens et al. (1997) constructed a hierarchical solution procedure (HSP) to buck a whole stand of northern hardwood stems into shorter logs in such a way that the optimal volumetric percentages for each log length-grade combination would be satisfied. The hierarchical optimization system was implemented as a two-stage model similar to the models of Eng and Daellenbach (1985), Eng et al. (1986), as well as Mendoza and Bare (1986). The model consists of an LP model at the upper level and a DP model at the lower level that are linked together through information exchange. The model also approaches the overall optimum iteratively, proceeding from one solution to another until the termination criteria are met. The upper level of the HSP model (i.e., the LP model), however, rather than passing on the shadow price of each log length-grade combination (log type) to the lower level, determines the price of each log type to be used at the lower level by the individual tree problem (ITP) procedure. Given this LP-created price vector, some number of additional new price vectors are generated by adding and subtracting a small amount to and from the original price of each log type. The lower level DP problem (ITP problem) is then solved separately for each new price set (price vector) created and the resulting log volumes are compared to those derived from customer orders. An ITP solution that satisfies all volumetric demand restrictions for all length-grade combinations is the optimum. If no such solution can be found, the search procedure then continues by including all these new price sets as new decision variables in the upper-level LP model, which is then re-solved to produce a new single price set for solving the lower-level DP problem. The optimal price set is thus a combination of one or more price sets, and the LP solution specifies the weights of each.

Heuristic approaches to stand-level bucking optimization have been offered by Laroze and Greber (1997) and Sessions et al. (1989), among others. Rather than trying to assign each stem class (diameter class) a bucking pattern or a weighted set of bucking patterns that maximizes the aggregate production value at the stand level, Laroze and Greber (1997) developed a Tabu Search (TS) based system for generating a set of bucking rules, one for each log type. A bucking rule comprises a log priority list (for details, see page 18 in the previous section) and three key attributes for each log type: (1) the minimum small-end diameter; (2) the quality classes of tree stems compatible with the log type; and (3) the maximum number of logs of that type that can be cut from each stem. The TS heuristic is used to explore the very large space of different rule sets (i.e., log-type attribute combinations). Given the volumetric demand constraints for the minimum proportion of long logs, the maximum proportion of short logs and the minimum average SED, as well as the original market price and price adjustment factor for each log type, the TS system iteratively searches for a bucking rule satisfying the market constraints while simultaneously maximizing the unit profit ($/ha). Each bucking rule generated by the TS system is evaluated against the given market constraints by cutting each class-representative tree using the stepwise bucking heuristics developed by the same authors (see the last paragraph in the previous section). This actually results in an optimal set of bucking patterns, with one distinct pattern generated for each stem class.

Sessions et al. (1989) applied a simple interval-halving binary search technique to find an appropriate price multiplier for long logs such that the given minimum ratio of the volume of long logs to the total log volume is achieved. As usual, the overall objective in their approach is to cut each tree length into log lengths so as to maximize the net value of the whole stand. The search process, initiated by the original, unadjusted set of log prices, first bucks each stem in a sample collected from the stand. The bucking itself is carried out using the network-based DP algorithm (Sessions 1988). If the resulting percentage of the volume in long logs is below the desired level, the prices for long logs are raised, and each tree is then bucked again using these new adjusted log prices. This two-sequence process continues until a price set is found that produces the desired proportion of long logs.

1.1.2.3 Forest level

The goal in forest-level bucking optimization is to assign each stand a bucking policy such that the overall production value from all stands to be harvested during a planning period will be maximized. This means that the stem-level bucking optimization procedure, in determining an appropriate bucking pattern for a single tree stem, should consider the log production potential not only of this particular stand but all the other stands included in the optimization process. On the other hand, because forest stands often differ markedly from one another in terms of species mixture, stand area, stand density (stems/ha) and, above all, individual tree characteristics (height, diameter at breast height, taper, quality, etc.), it may be inappropriate to cut each stand using the same bucking instructions and log product range (Arce et al. 2002). This is because a large number of products cut from the stand usually increases the time taken in sorting, loading and transportation operations, and thus the overall production cost. To achieve the best possible outcome at the forest level may thus require that in each stand only those log products be cut that are most compatible with the composition and characteristics of that particular stand. To summarize, the question in forest-level bucking optimization is about determining not only an optimal set of bucking patterns for each stand, but also an optimal allocation of products between various stands (i.e., which products and in what quantities should be produced from each stand).

The forest-level bucking optimization, compared to bucking optimization at the stem and stand level, has been studied and modeled much less. It seems that in recent years this important topic has been thoroughly addressed only by Laroze (1999) and Arce et al. (2002).

Laroze (1999) has proposed two models for forest-level bucking optimization, both being based on stand-level optimization models. One is an extension of the TS heuristic developed by Laroze and Greber (1997), while the other is an extended version of the price-directed two-stage LP/DP procedure originally proposed by Eng and Daellenbach (1985), Eng et al. (1986) and Mendoza and Bare (1986). Laroze calls this latter two-stage forest-level optimization method an LP/SP method because its stem-level bucking optimization is carried out using the shortest-path (SP) node labeling algorithm (see Sessions et al. 1989) rather than the conventional DP approach. In the forest-level TS method (LP/TS for short), the first task is to generate some number of alternative merchandising restriction sets, each specifying a minimum average small-end diameter, a minimum volumetric proportion of long logs, and a maximum volumetric proportion of short logs (i.e., the same three key attributes as used in Laroze and Greber (1997) to comprise the bucking rules). Given a set of merchandising restrictions and the stand descriptions, their TS method then generates a bucking rule for each log type in each stand. Finally, an LP model is used to break down the area of each stand between the various bucking rules generated in such a way that the overall net profit from all stands to be cut will be maximized. Rather than an optimal mix of stand and log-type specific bucking rules, the LP/SP method seeks a combination of bucking patterns for each stem-size class in each stand that yields the maximum total net profit at the forest level. Technically, the two-stage iterative LP/SP search method is identical to the stand-level LP/DP procedure in all respects other than the decision variable; that is, while the stand-level LP/DP model decision variable defines the number of stems of class j bucked using pattern k, the decision variable in the LP/SP method is the number of stems in stand i, class j, bucked using pattern k.

The third two-stage hierarchical system for the forest-level bucking optimization developed by Arce et al. (2002) in Brazil clearly differs from the LP/SP and LP/TS methods. First of all, Arce et al. explicitly integrated log product allocation and log bucking optimization into the same model. Their model evaluates both the different bucking pattern sets for each stem-size class in each stand and the different log product sets for each stand, and assigns each stand a mix of products and bucking patterns yielding the maximum total net revenue at the forest level. Second, the model of Arce et al. also takes into account the product-specific transportation costs from each stand to each mill in making decisions on the optimal allocation and bucking program for a given set of forest stands. This is important because, in general, transporting small amounts of wood from stands located far away from potential processing plants may be highly uneconomic and should thus be avoided. Third, Arce et al. formulated the upper level of their model as a mixed integer linear programming problem (MIP) and the cutting patterns for the MIP are generated at the lower level through a simple heuristic bucking procedure rather than a DP or SP algorithm.

1.1.3 Optimization on modern cut-to-length harvesters

The first cut-to-length (CTL) harvesters employing the bucking-to-value optimization method were introduced in Sweden during the second half of the 1980s (Sondell 1987, Nilsson and Sondell 1987). Quite soon it appeared that harvesters applying bucking-to-value optimization had a strong tendency to produce log output distributions with high proportions of both small and short logs and also large and long logs (Bergstrand 1990). At most Swedish sawmills, however, the desired log output distributions represented relatively even length-diameter distributions (i.e., each length-diameter combination showed almost the same target proportion). It followed that the typical bi-modal log output distributions from harvesters were poorly adapted to customer-oriented sawntimber production.

At the beginning of the 1990s, Bergstrand (1990) proposed that the obvious conflict between the stem-level and stand-level bucking optimization might be resolved by incorporating both the log values and the desired log output distributions into the bucking optimization system on harvesters. This new bucking principle, commonly called bucking-to-order optimization, apportionment bucking or dimension-apportionment merchandising (von Essen and Möller 1997b, Möller et al. 2002), requires that each log product (i.e., timber assortments like Scots pine sawlog, Norway spruce veneer log, etc.) be assigned two matrices: a value matrix (also known as a price matrix or a price list) and a demand (target) matrix. The value matrix of a particular log product specifies how valuable or profitable it is to cut different length-diameter combinations of this log type, while the corresponding target matrix specifies how desirable their cutting is from the customer's (e.g., the sawmill's) viewpoint. In essence, the main idea is to compromise between the stem-level and stand-level optimization. In order to improve the fit between the actual log output and log demand distributions at the stand level, it is necessary to relax the original aim of maximizing the overall value of each stem to some extent.

Bucking-to-order optimization can either be implemented through the adaptive price list method or the close-to-optimal method. In both these approaches, a harvester continuously monitors how far the actual cumulative output distribution of each log product is from the corresponding demand distribution. The bucking itself, however, still occurs on the basis of bucking-to-value optimization.

In the adaptive price list approach, the harvester calculates the difference between the actual and desired proportion of logs in each length-diameter class after cutting each stem and then adjusts the log values to be used to cut the next one using this information (Coggman and Gustafsson 1985, Bergstrand 1990, Ahonen and Lemmetty 1995, von Essen and Möller 1997a, Vuorenpää et al. 1997). Although the practical implementation of the adaptive price list method may vary between harvester models, the basic logic behind their price adjustment procedures should be the same: (1) the larger the difference between the desired and the actual proportions, the larger the change in log values; and (2) the value for the length-diameter combination with a surplus of material will be lowered and vice versa. In addition, in order to prevent a harvester from making inappropriate cutting decisions (e.g., cutting pulpwood logs from a stem section suitable for sawlog production), log values may usually be adjusted only within a certain price range (e.g., ± 5% from the original log value) definable by a harvester operator via a so-called adaptation factor.

In the close-to-optimal method, a harvester, instead of manipulating log values, keeps track of the desirability (priority) of each length-diameter combination within each log product, based on continuous comparison between the demand and actual output distributions of logs (Bergstrand 1989, 1990, Ahonen and Lemmetty 1995, von Essen and Möller 1997a, Vuorenpää et al. 1999). In the normal implementation, the harvester first generates a number of alternative bucking patterns for each stem: (1) the bucking pattern attaining the maximum total stem value (i.e., the bucking pattern suggested by the pure bucking-to-value optimization); and (2) all the bucking patterns with total stem values not deviating more than the given maximum (say, 5%) from the value of the solution of the bucking-to-value optimization. Given the product-specific priority matrices (priority tables), the harvester then calculates the total desirability value (priority value) of each bucking pattern generated at the previous phase. In the final step, the harvester cuts the tree into logs following the bucking pattern with the highest total desirability value (priority value) and, on completing the cutting procedure, updates all the relevant priority matrices.

Most CTL harvesters currently on the market apply the close-to-optimal method for fitting the log output distributions to the mill’s demand distributions (Sondell et al. 2001, Möller et al. 2002). Two recent studies (Möller et al. (2002, von Essen and Möller 1997a), however, clearly show that in real-life harvesting operations there are no large performance differences between the adaptive price list method and the close-to-optimal method in regard to the maximum fit achieved between the log output and log demand distributions. The adaptive price list method, however, seems to reach this maximum goodness-of-fit level somewhat more slowly than the close-to-optimal method. This is probably because the close-to-optimal method immediately starts cutting logs of a high desirability (priority) because at the beginning of the harvesting process the cumulative output matrices are empty and thus the priority matrices equal the original demand matrices. For the same reason, the adaptive price list method cannot adjust the log prices right after the cutting of the first tree stem, but has to wait until a sufficient number of logs have been accumulated in the output matrix of each log product. This may require cutting several dozen tree stems, depending on the stand structure and tree characteristics.

1.1.4 Optimization of value and demand matrices

Forest stands in typical Finnish conditions quite often differ considerably from one another in terms of stand structure and the characteristics of individual trees (see also Laroze 1999). This is true even if stands are growing in similar climatic and topographic conditions, and even if of the same biological age, site type, developmental stage (e.g., a young thinning, an advanced thinning or a mature stand), species mixture and size in area. Tree density (stems/ha) and the spatial pattern of trees (i.e., how they are distributed over the whole stand area), for example, may vary greatly from stand to stand. Similarly, the height, diameter and quality distributions of trees are usually more or less stand specific. The value (price) and demand (target) matrices, on the other hand are control tools; that is, their task is to affect a harvester's bucking process in such a way that the final bucking outcome matches both the mill demands and the forest owners' interests as close as possible. Apparently, because stands may be different in many respects, the same control action may not be equally efficient for all stands: i.e., a matrix combination performing well in one stand may not do the same in another stand (von Essen and Möller 1997a). An obvious question is whether we could improve the bucking outcome by adjusting the value and demand matrices prior to the actual harvesting operation.

So far, pre-control of value and/or demand matrices has been addressed in few studies, mainly for the following reasons.

First, although the fully mechanized cut-to-length harvesting system has gained ground worldwide, timber harvesting in many countries is still carried out by the tree-length and full-tree methods (Pulkki 1997, Godin 2001, Greene et al. 2001). Accordingly, the concepts of price and demand matrices as well as the adaptive price list and close-to-optimal techniques may be relatively unfamiliar to many operating in the field of bucking optimization although the bucking-to-value principle, for example, is a widely-known and widely-used optimization technique. Furthermore, in North America, for example, it seems to be more normal to specify the target numbers or volumes (proportional or absolute) for log lengths, rather than for each feasible diameter-length(-quality) combination of logs separately (e.g., Sessions et al. 1989, Pickens et al. 1997, Murphy et al. 2004). This tradition, however, may be slowly changing as more focus is being put on maximizing value recovery (Coyner 2004).

Second, in Sweden, where the bucking optimization systems for CTL harvesters were originally developed in the early 1990s, price matrices (or price lists as they are called in Sweden) controlling the bucking process on harvesters cannot freely be altered while harvesting. This is simply because the matrices agreed in the negotiations between the landowner and the forest industry representative actually determine the amount of money paid for logs of various sizes and qualities. In Sweden, a harvester thus seeks to assign each tree stem a bucking pattern maximizing the forest owner's sales income. As already stated, since a bucking policy of this kind may, however, not result in a log outcome optimal for the demands of customer orders, a more flexible bucking-to-order optimization principle was developed that considers both forest owner's and forest industry's interests.

In Finland, on the other hand, all logs within the same product (e.g., Scots pine sawlogs, Norway spruce veneer logs, etc.) share the same unit price per volume (€/m^3) whatever their physical dimensions are; some premium is usually being paid for logs of the highest quality though. What is even more important is that these timber market prices paid to Finnish forest owners need not have anything to do with the individual log values of the corresponding price matrices; that is, the stumpage prices do not actually control the bucking process on harvesters as is the case in Sweden. Consequently, one may freely assign each log product an initial price matrix and make further changes to it while harvesting to help achieve the desired log output distribution. A forest owner’s interest is safeguarded by converting each distinct stem section into logs of the highest value product possible (i.e., a stem section available for sawlog production, for example, is fully exploited as sawlogs up to the point where the stem diameter equals the SED of that particular sawlog product).

Despite an opportunity to generate stand-specific price matrices, the standard practice in Finland has been to cut all stands allocated for harvesting within the same time horizon under the control of the same price matrix set. The common view has been that no pre-control of price matrices is needed, because the on-line bucking-to-order procedure accommodates the log output distributions to the desired ones. This view, however, is mainly based on the results of bucking simulations using a few intuitively generated price matrix candidates within few stands (e.g., Vuorenpää et al. 1997) and has never been tested properly.

Similarly, the pre-control of the overall demand matrix of each log product into stand-specific sub-targets has been considered unnecessary. Imponen (2001a), for example, states that the only thing that matters is the forest-level fit between the overall log demand distribution and the actual cumulative log output distribution. Thus, although the use of the same demand matrix may result in a poor stand-level fit between the demand and output distributions, the overall fit at the forest level may still be quite good. This is because stands of different sizes, ages and structures are likely to produce different log output distributions which, when combined together, may provide a good match to the overall log demand distribution. However, this is not to say that allocating the overall demand into stand-specific sub-targets would not make the fit between the demand and output distributions at the forest level any better, or get the same fit at a better value.

Näsberg (1985) demonstrates that, when bucking on harvesters is controlled by the price matrices only, achieving the target log output distribution usually requires using more than one price matrix per product. Näsberg formulated his penalty-based approach to finding an optimal price matrix set as a goal interval programming model (GIP) and solved it using the same iterative Dantzig-Wolfe column generation – decomposition technique as did Eng and Daellenbach (1985), Eng et al. (1986) and Mendoza and Bare (1986). That is, each iteration cycle in Näsberg’s model consists of three interrelated steps. (1) Given the prices of all feasible diameter-length combinations of logs in a matrix form, determine an optimal bucking pattern for each stem class (actually for a representative tree in each stem class) using the longest route bucking algorithm. (2) Check to see if any of the bucking patterns generated in step 1 can help in achieving the desired log output distribution. This is done by solving the dual problem to the restricted version of the original upper-level GIP problem (a so-called RMP problem), the solution providing the marginal value for the increase in the number of trees in each stem class. If the value of the optimal bucking pattern from step (1) exceeds the marginal value of the corresponding stem class, this bucking pattern is then introduced into the RMP as a new column. (3) Solve the RMP problem with the new bucking patterns added from step (2) and check whether the resulting log output distribution matches the demand distribution perfectly. If not, first determine a new price for each log class: (a) in the case of a log surplus, subtract the marginal cost for additional logs from the original price; and (b) in the case of a log shortage, add a marginal value for additional logs to the original price. The marginal values and costs for various log classes come from the solution to the dual of the RMP problem. Using these new log class values, determine an optimal bucking pattern for each stem class (i.e., go back to step (1)). The optimal solution from this iterative DP-GIP procedure defines the number of times each bucking pattern is applied to each stem class (i.e., how many trees in each stem class should be bucked using a particular cutting pattern). Because a bucking pattern generated for each stem class at each iteration is the result of applying a particular price matrix to value bucking all trees in a given tree population, the optimal solution actually defines the frequency with which each price matrix generated should be used for each stem class.

Näsberg's optimization approach (1985) is advantageous in the sense that it provides an optimal solution which is operationally straightforward to implement on modern CTL harvesters. The main problem associated with this approach is that it requires that all trees be classified into stem classes defined by DBH and further assumes that all trees in the same DBH class are of the same size and taper. While probably valid in some plantation forests, this assumption may not be true in typical Nordic conditions, as can be seen in Fig. 4. Thus, as Näsberg admits, a bucking pattern which is optimal for a tree stem selected for a representative tree in a certain stem class may be highly sub-optimal for other trees in the same stem class. If stem classes were defined not only by DBH but also by tree height and quality, for example, the homogeneity within each stem class would probably be much better. This would result in at least near-optimal bucking patterns for all trees in the same stem class. But as Näsberg and Pickens et al. (1997) state, designing such a multidimensional stem classification scheme is difficult for both softwoods and hardwoods. Even if we managed it, how is it possible – usually without any detailed measurement data on the tree characteristics – to identify the correct stem class for each tree to be harvested. Furthermore, increasing the number of potential stem classes inevitably increases the number of decision variables in the optimization model and thus the computational burden of the model.

It should be stressed that Näsberg (1985) only applied the bucking-to-value procedure when converting trees into log lengths. His results thus do not show whether there would have been any need to use several price matrices per log product if the log conversion had been carried out using the bucking-to-order procedure. In addition, Näsberg's approach is clearly a one-product-one-stand optimization model. Neither does Näsberg address the forest-level allocation of the overall demand matrix into stand-specific sub-demands, because his optimization approach operates at the stand level only. Thus, although excellent in many respects, Näsberg's pioneering work in the field of price matrix optimization cannot provide an exhaustive answer to the question asked at the beginning of this section. That is, could we achieve better log output distributions if we cut each stand using stand-specific price and/or demand matrices rather than standard, non-stand-specific matrices?

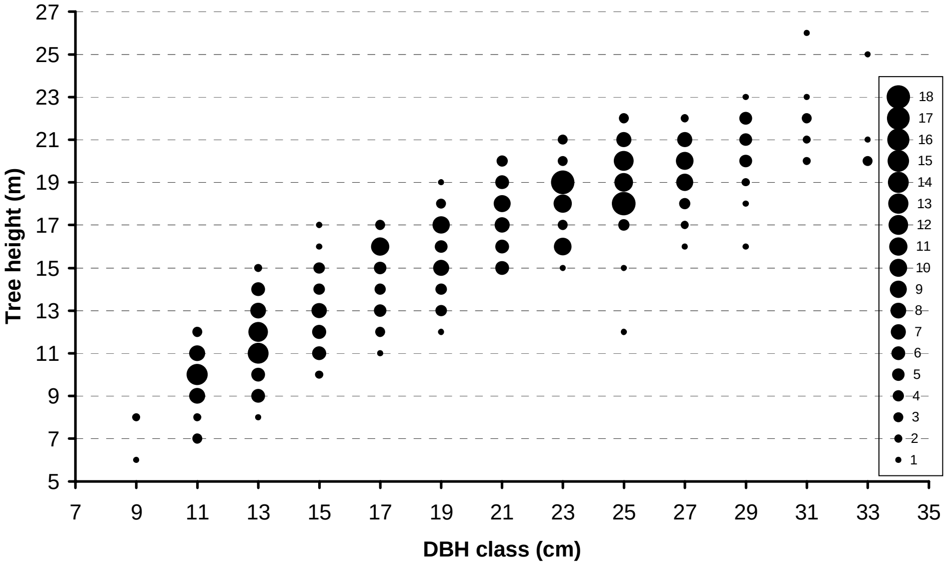

Figure 4. The tree height distribution by DBH class for 407 Norway spruce (*Picea abies* (L.) Karst.) trees from a mature spruce-dominated stand of 1.2 ha in southern Finland (stand no. 6 in Uusitalo (1997)). The size of the dot indicates the number of trees represented by each point, the legend at the right hand side showing the correspondence between the various dot sizes and numbers of trees.

Controlling the wood flow from forests to mills in such a way that each mill gets the desired log products in desired quantities and qualities at desired times has recently been seen as an even more important area in wood procurement development than the traditional work to reduce transport and harvesting costs. The field tests with modern bucking-to-order harvester systems, for example, have shown that a revenue increase of up to 10% can be achieved provided that the following prerequisites are met: (1) accurate and reliable pre-harvest information on both forest stands and customer orders is available; (2) stands, in terms of their composition, are well suited to the needs of customer orders; and (3) advanced logistics and communication systems are available (Sondell and Mitchell 2004).

Imponen (2001b) has estimated that improving the fit between the log demand and actual log output distributions at Finnish sawmills by only 5% might contribute additional revenue of €1…€2/m^3 of sawlogs. The annual production of sawn timber in Finland is approximately 10 to 12 million m^3. Thus, assuming a cubic recovery ratio (CRR) of 0.5 (i.e., 2 m^3 sawlogs are needed to produce 1 m^3 sawn timber), a better match between the log demand and log output distributions could provide an additional yield of 20…50 million Euros for the Finnish sawmill industry per year.

1.2 Study framework, objectives and limitations

The starting point in this thesis work is that we have an optimal overall log output distribution available for each given log product. Such an optimal demand matrix can be derived from the end product orders, the sales forecasts for various products, the stocks of the manufactured products at the mill, and the raw material supplies available at the mill and in the forest through the advanced production planning systems such as those described in Usenius (1986, 1999a, 1999b) (Fig. 5). These optimization calculations are usually done in close co-operation between the mill and the timber supply unit. The latter is then responsible for controlling the wood flow from individual stands, with the target being not only to provide the mills with a sufficient volume of wood raw material but also to match that to the desired log output distributions. Even if we ignore the log product allocation between stands, assuming that each given log product is cut in each stand, we can still decide on what kind of value and demand matrices we use on harvesters to control the bucking of tree stems in each stand.

The primary objective of the research presented in this thesis was to test the hypothesis that price and demand matrices adjusted to the unique conditions of each individual stand will perform better than standard, non-localized price and demand matrices. The performance criterion used was the physical fit between the original log demand distribution(s) from a mill and the actual log output distribution(s) from a harvester.

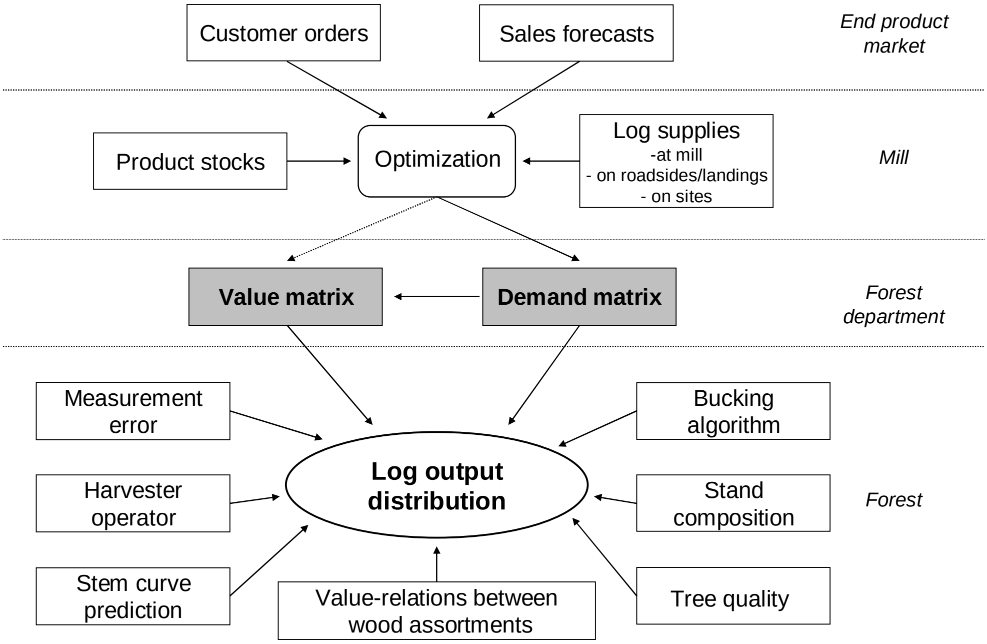

Figure 5. The framework of the thesis. Of the factors affecting the bucking outcome of a modern CTL harvester, only those in the two highlighted boxes will be covered. Figure based on Uusitalo and Kivinen (2001).

Testing this hypothesis requires that we have access to a control/calibration system. This system must carry out the search for well-performing stand and product specific price and/or demand matrices, given the overall log demand distribution for each log product and the stem profiles in each stand to be cut. Because such search tools are largely absent, the secondary objective of this thesis work was to devise an easy-to-implement search system for both stand-level and forest-level matrix optimization.

The third objective was to discover appropriate fitness metrics for the fit between the log demand and log output distributions at both stand and forest levels, and to evaluate their advantages and disadvantages in relation to the requirements for an ideal fitness measure.

The thesis consists of four studies, the first three addressing the first two objectives (Table 1) and the fourth focusing on the third objective. In Study I, a fuzzy control (FC) system was developed to establish a stand-specific price matrix for one log product in one stand at a time (the one-product-one-stand approach). Likewise, Study II focused on comparing the performance of stand-specific price matrices to that of uncontrolled reference price matrices, but with the price matrices of a given log product precontrolled in parallel at the forest level through a genetic algorithm (GA) based system (the one-product-several-stands approach). The modified version of this GA system was further employed in Study III to incorporate the overall log demand distributions into stand-specific sub-demands for all stands and all products simultaneously (the several-products-several-stands approach).

Table 1. The main features of the optimization approaches addressed in Studies I-III.

Model features	Study I	Study II	Study III
Optimization level	stand level	forest level	forest level
Control target	price matrices	price matrices	demand matrices
Number of log products included in optimization	1	1	2
Control technique	fuzzy control	genetic algorithm	genetic algorithm

The performance of the various price matrix sets, demand matrix sets and goodness-of-fit measures was tested virtually using bucking simulators rather than real harvesters in the actual forest environment for two reasons: (1) stand-specific price and demand matrices, if designed poorly, might have resulted in a large reduction in the volumes of high-value products and thus large economic losses for both the timber buyer and seller; and (2) it is hard to find several stands with precisely identical composition. All the simulations in the four studies focused on the bucking of Norway spruce (*Picea abies* (L.) Karst.) tree stems only and were carried out under the following assumptions (cf. Fig. 5):

- Trees are fault free, thus making it possible to apply fully automatic bucking (see Uusitalo et al. 2004) while harvesting
- The harvester is able to measure the profile of each tree without any errors in length and diameter values
- The harvester makes no errors in predicting the profile for the unknown part of a tree stem
- The measurements by a harvester and mill measurement systems do not differ (i.e., the harvester and the mill systems perform equally accurately in measuring log diameters and lengths).

This work is carried out from the viewpoint of the forest industry. A price or demand matrix set that yields a log output distribution closely matching the one desired by industry is considered as a good solution regardless of the effects it may have on the volumes of log products harvested in stands and the forest owners' harvesting income.

2 BRIEF INTRODUCTION TO FUZZY CONTROL AND GENETIC ALGORITHMS

2.1 Fuzzy control

A classic control approach is to construct a precise mathematical model of the system or process to be controlled. Then, given the desired output (the requested state) and the actual measured output (the observed state) of the system being controlled, we can assign input variables with values that bring the future state of the system closer to the desired state. Many real-world processes and systems are, however, often so complex, time-varying, full of non-linearities, and/or hit by unpredictable external disturbances that modeling them precisely in a mathematical form is difficult if not impossible (Klir and Yuan 1995, Puolakka 1997). A poor mathematical model, on the other hand, may easily result in erroneous control actions, and produce a large deviation between the desired and actual system states.

Fuzzy control (also termed fuzzy logic control, FLC) is an expert system based on fuzzy logic (Puolakka 1997). Fuzzy logic itself is founded on the fuzzy subset theory first introduced by Zadeh (1965). Unlike the classic set theory, fuzzy set theory also allows for partial membership of a set; that is, an element of a given universal set X (the universe of discourse) may have full membership, full non-membership or partial membership of a given subset A. These degrees of membership are usually determined by a membership function (μ_A) which assigns each element in the universal set a real number within the interval [0,1], the value of 1 indicating complete membership and 0 complete non-membership, and the values in between intermediate degrees of membership (the closer to 1, the stronger the membership). Similarly, fuzzy logic permits partial truth values rather than only completely true and completely false as is the case in conventional Boolean logic.

A fuzzy expert system, commonly called a fuzzy controller, replaces the mathematical model responsible for deriving appropriate control actions in a classic control system. The system inputs and outputs, however, usually remain unchanged. A fuzzy controller typically consists of four interrelated modules: a fuzzification module, a knowledge base (a rule base), a fuzzy inference engine and a defuzzification module.

Prior to the actual fuzzification process, the feasible value range of each input variable of the fuzzy controller needs to be divided into a few (say, 3-7) equally or unequally sized sub-ranges (intervals), each labeled with an appropriate linguistic term (e.g., approximately zero, large positive, very hot, slightly negative and so on). Each linguistic state of each input variable is then represented by an appropriate membership function; that is, each linguistic state defined for a given input variable actually represents a fuzzy set, the variable itself being usually called a linguistic variable. A membership function curve can take any shape, the only condition being that its values must vary between 0 and 1, inclusive. The most widely used shapes include the triangle, trapezoid, bell-shape (Gaussian distribution curve) and sigmoid curves. The selection between these curves is primarily based on the nature of the control problem to be solved.

The fuzzification sub-process converts the precise measurement value of each input variable into the degrees of membership of all fuzzy sets defined on that particular input variable (Fig. 6). That is, fuzzification is a mapping from a set of real numbers (the range of possible values of an input variable) depicting the state of the process being controlled, to membership values (degrees of membership) between 0 and 1.

The inference engine performs reasoning about the control actions required to alter the current state of the system/process to the desired one. The reasoning in a fuzzy controller is based on a knowledge base (a fuzzy rule base) and the fuzzified input values from the preceding fuzzification sub-process. The knowledge base contains the relevant control knowledge in the form of conditional if-then rules:

Rule 1: IF x is A_1 AND y is B_1 THEN z is C_1
Rule 2: IF x is A_2 AND y is B_2 THEN z is C_2
…
Rule n: IF x is A_n AND y is B_n THEN z is C_n

Where x and y are input variables of the fuzzy controller, z is an output variable (i.e., represents the control action to be taken), and A_i, B_i, C_i are fuzzy sets defined on x, y and z, respectively.

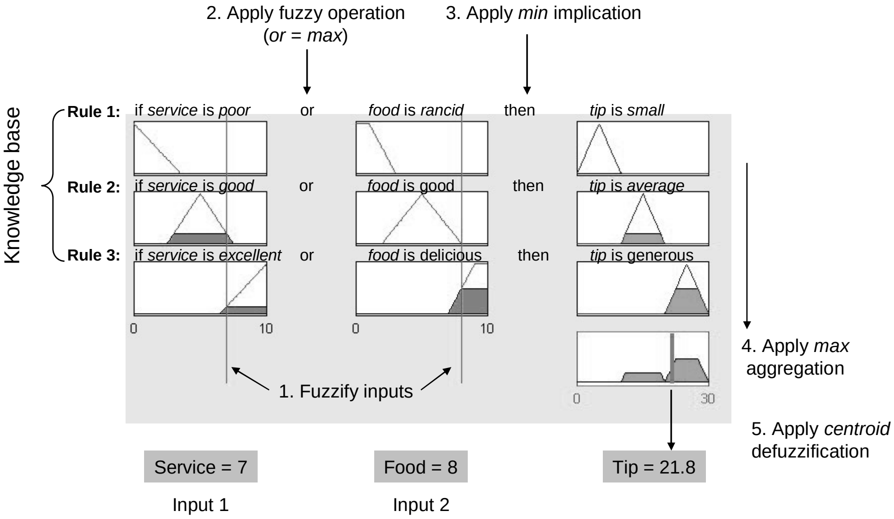
2. Apply fuzzy operation (*or* = *max*)
3. Apply *min* implication
Knowledge base
Rule 1: if *service* is *poor* or *food* is *rancid* then *tip* is *small*
Rule 2: if *service* is *good* or *food* is good then *tip* is *average*
Rule 3: if *service* is *excellent* or *food* is delicious then *tip* is generous
0
10
0
10
0
30
4. Apply *max* aggregation
5. Apply *centroid* defuzzification
1. Fuzzify inputs
Service = 7
Food = 8
Tip = 21.8
Input 1
Input 2

Figure 6. The fuzzy inference process exemplified by a two-input one-output three-rule tipping problem (Fuzzy logic toolbox… 1998). The quality of both the service and the food at a restaurant is rated on a scale of 0 to 10 (0 = very poor; 10 = excellent), the amount of the tip varying from 0 to 30%. Three fuzzy subsets are defined on both the two input variables (poor/rancid, good, excellent/delicious) and the single output variable (small, average, generous).

The inference process actually consists of three sub-processes (Fig. 6). First, the inference process determines the truth value of the premise part (the antecedent) of each rule (i.e., to what degree the rule applies) using the membership values from the fuzzification sub-process and classical or customized fuzzy operators for the logical operations AND, OR and NOT. For example, the logical operations AND and OR are classically resolved using the min and max functions, respectively. The second task is to calculate the degree to which the consequent part of each rule applies. This is usually done by applying either the min or product inferencing method; the former truncates the resulting fuzzy output set at a height corresponding to the truth value of the rule's premise, while the latter scales the fuzzy output set. In the third phase, all the fuzzy output sets assigned to each output variable are aggregated into a single fuzzy set. The aggregation can be done by taking either the pointwise maximum (max composition) or the pointwise sum (sum composition) over the whole fuzzy set assigned to the output variable by the two preceding subprocesses. Thus, the inference process is often referred to by such terms as max-min inference and sum-product inference.

The output of a fuzzy controller always comprises one or more fuzzy sets (as many as there are control variables). However, a fuzzy set can seldom be used as a direct input into the system/process being controlled but needs to be converted into a single crisp value. This conversion is done by the defuzzification sub-process (Fig. 6).

2.2 Genetic algorithms

Genetic algorithms (GAs) are probabilistic algorithms based on the mechanisms of biological evolution and used for solving complex search and optimization problems (Michalewicz 1996, Mitchell 1996). The appeal of using evolution as a framework for designing problem-solving methods arises from the apparent analogies between evolution and computational problems (Mitchell 1996). First, many computational problems in science and engineering require sifting through an enormous number of potential solutions. This is also precisely what occurs in nature where a huge number of sets of gene sequences are continually tested and changed by evolution. Second, conditions in nature change continually. Thus evolution is in effect seeking solutions in the face of constantly changing circumstances. Clearly, this is often the case with many computational problems. For example, the major challenge in robotics is to get a robot to perform well in a variable environment. Third, many scientific and practical problems in many disciplines are hard to solve because of their extreme complexity. While nature itself is a highly complex system, it has been created by surprisingly simple rules of evolution over millions of successive generations.

Genetic algorithms differ from the conventional search and optimization methods, such as hill climbing, Monte Carlo, random search and greedy methods, in two important respects. First, rather than operating on one solution at a time, GAs process a number of potential solutions in parallel; that is, GAs are population-based algorithms. Second, as in nature, better individuals to solve a given problem are generated by means of three genetic processes: natural selection, recombination (crossover) and mutation.

A typical genetic algorithm consists of an initialization phase and an evaluate-check for the termination-select-crossover-mutate loop. The pseudo-code below (adopted from Michalewicz 1996, Michalewicz and Fogel 2000) describes the basic structure of a GA:

```
Procedure genetic algorithm
    begin
    $t \leftarrow 0$
        initialize P($t$)
        evaluate P($t$)
        while (not termination-condition) do
            $t \leftarrow t+1$
            select P($t$) from P($t$-1)
            alter P($t$)
            evaluate P($t$)
        end
    end
```

where P(t) = $\left\{x_1^t,...,x_n^t\right\}$ is a solution population for iteration t.

In the initialization phase, an initial set of potential solutions is generated randomly, or if some problem-specific knowledge about the most promising search areas is available, some heuristics using this prior information can be applied (Alander 1998, Michalewicz 1996). Next, the performance of each solution candidate is evaluated (i.e., how good the solution is for the problem at hand). The evaluation function, generally termed a fitness function, may be derived directly from the problem itself (as is often the case with function optimization) or it can be a combination of several measures (each assessing the quality of the solutions on a particular criterion) (Beasley et al. 1993a). At this stage, the search process is halted if a predefined stopping criterion is reached. In most cases, the stopping rule simply defines the maximum number of iteration cycles (generations) the GA is planned to run. If the stopping criterion is not met, a new set of potential solutions is created on the basis of the current candidate solutions. At the start, some number of individuals (often equal to the GA population size) are selected from the current population according to a scheme which favors the fitter individuals. This is equivalent to the principle of natural selection, according to which the highly adapted organisms are more likely to survive in changing environmental conditions. In crossover, some of these newly selected individuals are combined with each other to form new solutions to a problem. In the final step, the new population produced by the operators of selection and crossover undergoes a random mutation in which each individual (real-valued representation) or each element in the solution chromosome (binary encoding) has the same small probability of being altered. In this way, the population of potential solutions evolves over successive generations, the goal being to converge toward the globally optimal solution.

Selection, crossover and mutation can all be implemented in many ways. One of the most widely used selection methods is fitness-proportionate selection (also called roulette-wheel selection), in which the probability of an individual being selected to the population of the next generation is proportional to its fitness value (Michalewicz and Fogel 2000, Mitchell 1996). Other selection methods include techniques such as elitism (a fixed proportion of the best individuals is always passed on to the next generation), rank selection (individuals are selected according to their ranks rather than their absolute fitness values), tournament selection (first select two individuals randomly from the population, then select the fitter as a member of the new population), and various “scaling” methods (select individuals according to their expected, “scaled”, values rather than their “raw” fitness values) (Mitchell 1996).

In the context of single-chromosome individuals encoded as classic binary strings (i.e., strings of 0s and 1s), the simplest form of crossover is a single-point crossover (Figure 7a). This type takes two individuals, chooses a crossover position at random, and finally swaps the tail parts of the two parents after the chosen position to form two new individuals (e.g., Beasley et al. 1993a). Two individuals represented as real numbers can be crossbred, for example, by using the average, geometric mean or the extension operator (Beasley et al. 1993b). When applied to individuals represented as bit strings, mutation in its simplest form randomly flips one or more of the bits in each chromosome string (Figure 7b). In real-value coded GAs, individuals can be mutated, for example, by replacing the present value by a random one (random replacement), by adding or subtracting a small, often randomly generated value from the present value (creep), or by multiplying the present value by a random amount close to 1 (geometric creep) (Beasley et al. 1993 b).

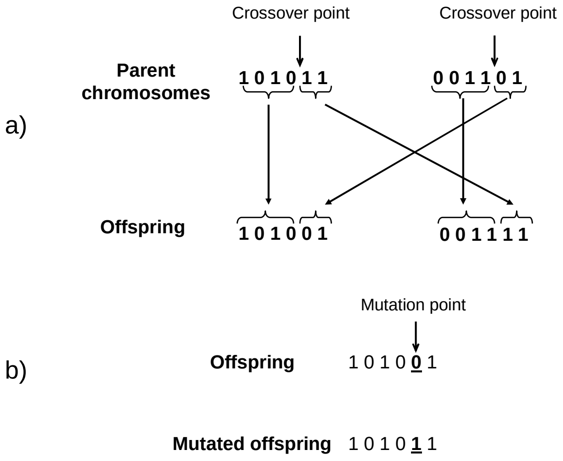

Figure 7. Single-point crossover (a) and single mutation (b).

The history of genetic algorithms dates back to the 1960s when the American researcher John Holland studied the mechanisms of natural adaptation in order to develop adaptive computer systems (Mitchell 1996). Holland's idea itself was not new but his implementation with a population-based search strategy employing random variation operators was a totally different approach in the field of evolutionary computation. Holland’s theory (Holland 1992) assumed that good solutions are made up of good building blocks (schemas or schemata). In GAs, individuals with higher fitness values (i.e., with good building blocks) are thus given more opportunity to reproduce than poorly performing ones. This means that seeking the optimal solution(s) focuses on the most promising areas in the search space. However, since selection alone is not able to introduce any new points within the search space, operators providing some random search capability are needed. In crossover, the idea is that recombining good solutions with each other (i.e., merging good building blocks together) may produce offspring with even higher fitness than that of their parents. The role of mutation, in turn, is to try to prevent premature convergence to a local optimum (i.e., mutation attempts to keep up the fitness variance in the population).

3 MATERIAL AND METHODS

3.1 Stands

Fifteen real-world and 10 virtual Norway spruce (*Picea abies* (L.) Karst.) stands were included in the study. The real forest stands were used in Studies I-III to construct stand-specific price and demand matrices and test their performance against the uncontrolled reference matrices. The virtual stands were created for evaluating the behavior and performance of the four goodness-of-fit measures in Study IV.

The 15 Norway spruce sites were selected as study stands subjectively from stands available for harvesting during the period from summer 1998 to autumn 2000. The primary aim in stand selection was to have a collection of stands showing wide variation in the DBH distributions of spruce trees. As a result, all the most typical DBH distribution shapes – normal with one or more peaks, uniform, left skewed, and right skewed – were represented in the study material (see Fig. 2 in Study III). Fourteen of the 15 stands were clear-felled, and one was a seed-tree cut. The total cutting area of the individual stands ranged from 0.8 to 4.8 ha and the total cutting volume of spruce logs from 191 to 685 m^3, with a mean stem size ranging from 302 to 864 dm^3. The total number of spruce stems in all 15 stands was 12 389, of which 9 155 stems were classified as non-pulpwood stems[1]. All stands were privately owned and located in southern Finland close to the cities of Lahti and Mänttä. A more detailed description of the characteristics of the 15 study stands can be found in Studies II and III.

Two types of stand-specific price and demand matrices were constructed for each stand included in Studies I-III: (1) matrices adjusted by the real stem data; and (2) matrices adjusted by the estimated stem data. Studies II and III included all 15 Norway spruce stands, while Study I included only 4 of them (only stands 10, 11, 12 and 13 were available at the time when the analyses of Study I were carried out).

The real stem data refers to the stem profiles measured and stored by a harvester while processing trees during the harvesting operation. This data was gathered in each stand by a Ponsse single-grip harvester which measured the stem diameters of each tree from the butt to the top and stored them at 10-cm intervals in one or more stem data files (STM files) following the joint Nordic StanForD standard for harvester data communication (Standard for Forest… 2004).

The estimated stem data refers to the stem profiles derived from the sampling data gathered during a cruise prior to harvesting. Each of the 15 study stands was inventoried by either the author or one of two other experienced forest professionals using the preharvest measurement method developed by Uusitalo (1997). The sampling data, typically consisting of 8 or 10 basal area measurements and roughly 30-50 sample trees measured for at least DBH, was analyzed using the EMO software package planned and programmed by Uusitalo and Kivinen (2000). EMO generates the DBH distribution for each tree species in a stand using the kernel smoothing technique, predicts the tree height distribution for each DBH class of each species using either Lappi’s (1991) or Näslund’s (1936) height model, applies the stem curve equations from Laasasenaho (1982) to create the tree profiles, and finally lists the profiles generated in a text file in STM format.

The 10 virtual Norway spruce stands for Study IV were created using the Weibull function (Bailey and Dell 1973), Näslund’s height model (1936) (parameters a and b taken from Uusitalo (1997)) and Laasasenaho’s stem curve equations for spruce (1982). Five ‘stand types’ were represented: (1) a stand with a normal DBH distribution and a small mean DBH; (2) a stand with a normal DBH distribution and a large mean DBH; (3) a stand with a uniform DBH distribution; (4) a stand with a right-skewed DBH distribution; and (5) a stand with a left-skewed DBH distribution. Two instances of each stand type were constructed: (1) a small stand with 380 spruce stems in total, and (2) a large stand with twice as many trees.

[1] Spruce stems were classified into the pulpwood stem category if the dbh was less than 180 mm.

3.2 Demand and price matrices

The overall demand matrices of the two Norway spruce log products (sawlogs and veneer logs) were either direct copies or slightly modified versions of the corresponding matrices used in everyday wood procurement at the time of data collection (Table 2). A slight modification relates to changing the original demand matrix such that the individual target values were specified separately for each diameter-length combination of logs rather than for each length class within each diameter class (i.e., a target by diameter classes is replaced with a so-called target over the entire matrix). This change also required that the individual cell entries be redefined by proportioning the original target values to the given cumulative target sum, either equaling 1 000 or 10 000.

The demand matrix of spruce sawlogs in Study I and one of the two demand matrices tested in Study IV (matrix T1) came from Koskitukki Oy, the wood procurement company of the medium-sized sawmilling company Koskisen Oy. The Study I demand matrix and the T1 matrix of Study IV were, however, not identical in size or in target values. The T2 demand matrix, the other demand matrix of spruce sawlogs in Study IV, was derived from matrix T1 by swapping two adjacent columns.

The demand matrix of spruce sawlogs in the two other studies was obtained from the Finnforest Group’s two sawmills; the Study II matrix from the Vilppula sawmill and the Study III matrix from the Kyröskoski sawmill in Hämeenkyrö. The demand matrix of spruce veneer logs in Study III was based on the matrix originally designed by Koskitukki Oy for Koskisen Oy’s plywood mill in summer 2003.

Similarly, the reference price matrices used to test the performance of the precontrolled stand-specific matrices were either direct copies or slightly modified versions of the price tables designed by either Koskitukki Oy or Metsäliitto Osuuskunta (Metsäliitto) for use with original versions of the overall demand matrices mentioned above (Table 2). The reference price matrices were either uniform matrices (note: in Study I these matrices are called even matrices) with all log class entries sharing the same base value (Study II) or more specifically defined matrices which the companies, in their long-term experiments, had found to perform well in most stands (Studies I and IV). The log values in the Study III price matrices used for testing the performance of stand-specific demand matrices were study-specified. The SED and log length classes were adopted from the corresponding company-designed price tables.

3.3 Bucking simulators

Four bucking simulators were involved in the price and demand matrix performance tests in Studies I-IV (Table 3). Three of them were different versions of the OptiSimu bucking simulator developed by the Ponsse harvester manufacturer, and one simulator (called VP-Simu) was programmed by the author. All the OptiSimu versions used employed the adaptive price list method to accommodate the log output distribution(s) to the desired one(s), while the VP-Simu simulator applied the close-to-optimal principle. None of the four computer programs, however, simulated the bucking process of a real harvester because they were not equipped with a taper prediction capability. That is, all the simulators assumed that after felling a tree, a harvester first delimbs and measures the whole tree length from the butt end to the top, then repositions the harvesting head at the butt end and starts bucking the tree.

Table 2. The main features of the overall demand matrices and the associated price matrices used in the performance tests in Studies I-IV.

Study	Matrix feature	Demand matrices		Price matrices		
		Spruce sawlogs	Spruce veneer logs	Spruce sawlogs	Spruce veneer logs	Spruce pulpwood logs
I	Origin	Koskisen sawmill	-	Koskitukki Oy	-	Koskitukki Oy
	Matrix size[a]	20 x 10	-	20 x 10	-	18 x 3
	Value range	0 - 272 (per 10 000 logs)	-	€0–52/m^3	-	€0–13/m^3
II	Origin	Finnforest's Vilppula sawmill	-	Metsäliitto	-	Metsäliitto
	Matrix size[a]	15 x 7	-	15 x 7	-	11 x 8
	Value range	0 – 21 (per 1000 logs)	-	Uniform matrix €50/m^3	-	Uniform matrix €17/m^3
III	Origin	Finnforest's Kyröskoski sawmill	Koskisen plywood mill	Metsäliitto	Koski-tukki Oy	Metsäliitto
	Matrix size[a]	14 x 8	12 x 5	14 x 8	12 x 5	8 x 3
	Value range	2 – 19 (per 1000 logs)	10 – 60 (per 1000 logs)	€50-51/m^3	€65-66/m^3	€10/m^3
IV	Origin	Koskisen sawmill	-	Koskitukki Oy	-	Koskitukki Oy
	Matrix size[a]	15 x 8	-	15 x 8	-	5 x 3
	Value range	0 – 30 (per 1000 logs)	-	€0-56/m^3	-	Uniform matrix €20/m^3

[a] number of SED classes x number of log length classes.

Table 3. Bucking simulators used in the performance tests in studies I-IV.

	Study			
	I	II	III	IV
Simulator	Ponsse OptiSimu	Ponsse OptiSimu	VP-Simu	Ponsse OptiSimu
Version	2.50	3.01.14	-	4.328
Manufacturer	Ponsse Oyj	Ponsse Oyj	Author	Ponsse Oyj
Bucking-to-order optimization method	Adaptive price list	Adaptive price list	Close-to-optimal	Adaptive price list

3.4 Control systems for generating stand-specific price and demand matrices

3.4.1 Fuzzy controller for stand-level control

In Study I, an iterative one-stand, one-product calibration system was constructed that accommodates the initial price matrix to the desired log output distribution given the stem profiles of one tree species. The accommodation is achieved through a closed control loop consisting of a fuzzy controller, implemented in the Matlab 5.2 environment using the Fuzzy Logic Toolbox (Fuzzy logic toolbox…1998), and the Ponsse OptiSimu bucking simulator. The fuzzy controller has two data inputs: (1) a deviation matrix declaring the relative difference between the actual output proportion from the bucking simulator and the proportion desired by a mill for each log class; and (2) the derivative of the deviation matrix showing the rate of change in the deviation (error) for each log class. Similarly, the output variable of the fuzzy controller is a matrix defining the relative change in the price of each log class (diameter-length category). The value ranges of both input variables and the output variable were divided into five linguistic states: negative large, negative, zero, positive and positive large. These states are expressed using either triangular or trapezoidal membership functions (fuzzy numbers).

Fuzzy reasoning about the appropriate price change in each log class matrix is based on the fuzzified values of the two input matrices and a fuzzy rule base consisting of 25 if-then implications and implemented using min inferencing and max composition. As a result, each log category is assigned a fuzzy output set that is then converted into a crisp output value (price change) by applying the popular centroid method which returns the centre of gravity for the area under the curve of the fuzzy output set (Klir and Yuan 1995). The resulting log-class-wise price changes are then applied to the present price matrix to produce a new one to be used by the bucking simulator for controlling the bucking process at the next iteration cycle.

This control loop was iterated 60-70 times until no significant progress in the fitness measure was observed. The fit between the log demand and actual log output distributions was measured using the traditional metric, commonly called the apportionment index (AI) or apportionment degree (outlined in detail in Study IV), first introduced in forestry by Bergstrand (1989).

The number of fuzzy sets defined for the two input and one output variables, and the shapes and locations of the associated membership functions, were determined by experimenting with the performance of several different fuzzy controller versions in one of the four study stands. The parameter combination that yielded the highest AI value during 60-70 iteration cycles in the test runs was selected for use in the actual performance tests.

3.4.2 GA systems for forest-level control

The two forest-level search systems for generating either stand-specific price matrices for one log product (Study II) or stand-specific demand matrices for several log products (Study III) were based on a genetic algorithm approach. The GA system in both studies comprises an actual GA module and an embedded bucking algorithm, both written in the C language. The bucking algorithm in Study II follows the bucking-to-value optimization principle and is implemented using Näsberg's forward reaching DP technique (1985). Study III, however, employs the bucking-to-order optimization to convert trees into logs, with the algorithmic implementation of the close-to-optimal method following Bergstrand's guidelines (1989).

The main body of the GA module was similar in Studies II and III. First, candidate solutions in both studies were encoded as real-valued price or demand matrix strings, with the number of matrices in each string being equal to the number of stands and log products included in the optimization. Thus, assuming a population size of *n* with *k* log products to be cut in each stand, the solution populations in Study II took the form of a two-dimensional table with *n* rows ($k = 1$). In Study III the solution populations were three-dimensional tables consisting of *n* rows, each row having *k* matrices for each stand. Second, the starting matrix population in both studies was initialized similarly, by assigning a random integer to each matrix entry (i.e., each log category) from a given value range. The crossover and mutation operators were also the same for the GA module in both studies. In the crossover, two matrix strings (candidate solutions) were selected randomly and combined stand- and product-wise together to form a new offspring matrix through a uniform crossover operator. In the mutation, the cell values in each matrix of each solution string were exposed to a random change. The crossover and mutation rates, however, were not the same for the two studies. In Study II, the crossover rate varied from 0.5 to 0.8 (depending on the degree of elitism used in the performance tests) and the mutation rate from 0.001 to 0.02. The corresponding rates in Study III were 0.9 and 0.01.

While having much in common, the GA modules of Studies II and III also had some distinct differences. First, the GA in Study II used elitism to perform selection while the GA in Study III applied tournament selection with a tournament size of two. The former method always selects a given fixed number of the best individuals to be passed on to the next generation. The latter method randomly selects two individuals from the present population, with the better (the one with the higher fitness value) being chosen to be a member of the next generation's population. Second, the fitness score assigned to each matrix string was calculated differently: Study II used the traditional apportionment index; and Study III used a fitness measure based on the statistical χ^2 goodness-of-fit test (hereafter referred to as the χ^2 measure). Both tests return a value indicating how closely the actual bucking outcome as a whole matches the desired outcome at the forest level; the higher the test value, the higher the overall similarity between the log demand distribution(s) and the cumulative log output distribution(s).

3.5 Experimental tests

Two tests with the developed fuzzy control system were conducted in **Study I**. The first focused on testing the system's ability to derive a known price matrix for each study stand. This testing was done by initiating the fuzzy system with an arbitrary price matrix (two different matrices were included in the test) and a demand matrix (produced by the known price matrix under the bucking-to-value optimization) and running the system 55-100 times. The comparison between the known price matrix and the two fuzzy controlled price matrices was carried out indirectly by calculating the apportionment index between the log distribution produced by the known price matrix and the log output distribution produced by the fuzzy controlled price matrices. The second test compared the performance of the fuzzy controlled price matrices against that of the reference price matrices. The price matrices were fuzzy controlled by both the stem data gathered by a harvester (real stem data) and the stem data compiled from the EMO measurements (estimated stem data). The performance measure was the apportionment index between the desired output matrix and the actual output matrix from the simulator. The bucking simulations were carried out under both the bucking-to-value and bucking-to-order optimizations.

Three different tests to evaluate the performance of the GA search system were carried out in **Study II**: (1) a test consisting of 10 separate runs with the same parameter set to see the variation in the system output (i.e., the best fitness value achieved during the iteration cycles); (2) a test exploring the effects of the population size, mutation rate and degree of elitism on the system output value, with each parameter effect tested at three levels; and (3) a test comparing the performance of GA-controlled price matrices to that of the uncontrolled reference price matrices. In the performance test, the GA system was first run for 1 000 iterations to assign each stand an optimum price matrix. The GA runs were done with both the harvester-collected and predicted stem data on the 15 study stands. Each stand was then cut with a bucking simulator (Table 3) using three different bucking strategies (see Table 3 in Study II): Strategy 1 used a uniform price matrix (a reference matrix) and an adaptation factor value of 20%; Strategy 2 used a GA-controlled price matrix (with the GA process allowing max. ±10% price variation from the uniform matrix price of 300) and an adaptation factor of 10%; and Strategy 3 used a GA-controlled price matrix (with the GA process allowing max. ±20% price variation from the uniform matrix price of 300) and the bucking-to-value optimization (i.e., adaptation factor = 0%). In all three bucking alternatives tested the demand matrix for spruce sawlogs was that used at FinnForest's Vilppula sawmill during the winter of 2000 (Table 2). The fit between the desired log output matrix at the Vilppula sawmill and the log output matrices from the simulations conducted was evaluated using the AI measure.

Study III included two tests: (1) a stochasticity test identical to test 1 in Study II; and (2) a performance test much similar to test 3 in Study II. In the first phase of the performance test, the GA system was run for 500 iterations with both the harvester-collected and EMO-generated stem data of the 15 study stands. Concerning the parameters of the close-to-optimal method, two different parameter settings were applied in the GA runs: (1) a setting referred to as 5%/10; and (2) a setting referred to as 20%/20. These settings mean that for each stem the best 10 or 20 bucking patterns with values not deviating more than 5 or 20% from the value of the bucking-to-value solution comprised the bucking pattern set from which the one with highest overall priority value was then selected for implementation. In the second phase, the bucking of each of the 15 study stands was simulated using both the bucking-to-value and bucking-to-order optimizations. The bucking-to-order simulations were carried out for both types of GA-controlled demand matrices (i.e., those adjusted by the real stem data and those adjusted by the estimated stem data) under the control of the same two parameter sets (5%/10 and 20%/20) as used in the GA runs. The bucking-to-value procedure was applied to the reference demand matrices only. Thus, seven different bucking simulations were carried out in each study stand. The fitness comparison between the overall log demand matrices and the cumulative log output matrices was done using the χ^2 measure.

Study IV was divided into two parts. The theoretical part first listed four essential characteristics required from an ideal goodness-of-fit measure for assessing the fit between the log demand and log output distributions. Four potential fitness measures were then introduced along with their mathematical representations: the apportionment index (also called apportionment degree), the χ^2 measure, Laspeyres' quantity index and the price-weighted apportionment index (degree). Finally, each measure was evaluated in relation to the requirements of the ideal fitness measure. The experimental part consisted of two simulation-based tests: one for exploring the behavior of the four measures in ten generic stands cut using two different price-demand matrix sets; and the other for analyzing the use of each of these measures as a decision criterion for choosing which of the two potential sawmills each stand should be directed to (i.e., which of the two demand matrix alternatives should be used for controlling the bucking in each stand). In this latter test, a stand was allocated to the sawmill whose demand matrix for spruce logs – according to the goodness-of-fit measure applied – best suited the stand. The resulting log output distribution in each stand was valued by log prices derived from the end product market through an advanced sawing planning system. These individual total log values were then added to assign an overall total log value to the chosen allocation policy. The reference value was obtained by cutting each stand according to the demand matrix alternative providing the higher total log value, and aggregating these values across all stands.

4 RESULTS

4.1 Control of price matrices at stand level (Study I)

The fuzzy control system did not manage to derive the desired price matrix from either of the two starting price matrices. When the search for the desired price matrix started from the uniform price matrix (all diameter-length entries with a non-zero target proportion had the same relative log price of 300), the average relative difference in log prices in the four test stands varied between 6.0 and 6.5%. When using the random price matrix as a starting-point for the search, the corresponding difference varied between 5.3 and 6.0%. It is thus quite understandable that the log output distributions generated by these fuzzy controlled price matrices did not perfectly match the log output distribution produced by the known price matrix in any of the four study stands. Still, the fit between the 'desired' output log distribution and the 'fuzzy controlled' log output distributions was rather good as the apportionment degree exceeded 92% in all study stands regardless of whether the fuzzy system was initiated by the uniform or random price matrix.

The effect of the pre-control of price matrices on the bucking outcome was clearly dependent on the adaptation factor employed in the bucking simulations (Fig. 8). According to the 3 x 3 factorial ANOVA (Table 4), the interaction between the price matrix and the adaptation factor was even statistically significant.

When the bucking simulations were carried out using the bucking-to-value optimization (i.e., the adaptation factor equaled 0%), the price matrices controlled with the harvester-collected stem data outperformed the Koskitukki base matrix ($p < 0.001$; pairwise comparisons of the means using the Tukey test). The performance difference between the Koskitukki base matrix and the price matrices adjusted using imperfect stem data was much smaller ($p > 0.2$).

Moving from bucking-to-value to bucking-to-order optimization improved the fit between the log demand and log output distributions for both the base price matrix and the price matrices pre-controlled by the pre-harvest inventory data (Fig. 8). This improvement was also statistically significant at the level of 0.001. No similar improvement in the apportionment degree occurred with price matrices adjusted with the harvester-collected stem data. In fact, with these matrices, doubling the adaptation factor from 10 to 20% produced a slight reduction in the goodness-of-fit values. At both the non-zero adaptation levels, the price matrices controlled by perfect stem data outperformed the base price matrix and the price matrices adjusted by the predicted stem profiles (Fig. 8). This outperformance also reached a statistical significance ($p < 0.001$ for the adaptation level of 10%, and $p < 0.05$ for the adaptation level of 20%).

It is important to note that the fuzzy system, while seeking the best price matrix for a given stand, was allowed to change the log prices by a max of ±20% from the prices of the base matrix. Thus, when the bucking simulator in bucking-to-order optimization was allowed to change log prices by another ±20%, the maximum change in price was actually as large as -36 or +44% from the original. This is to say that the simulation results obtained with the base price matrix at an adaptation level of 20%, for example, are not in fact comparable to the results obtained with the fuzzy controlled price matrices at the same adaptation level. Instead, because the changes made to the original log prices by the fuzzy system varied in all stands but one between -15.4 and +16.5%, the simulation results obtained with the fuzzy controlled matrices at an adaptation level of 10% should preferably be compared to the results obtained with the base matrix at an adaptation level of 20%. When the comparison is done this way, the price matrices controlled with the fuzzy system using the real stem data still performed better than the base matrix ($p < 0.05$), which in turn performed better than the matrices adjusted by the pre-harvest inventory data ($p > 0.07$).

The adjustment of log prices, whether done prior to or during the bucking simulation, resulted in a slight loss in the volume of spruce sawlogs with few exceptions. For example, when the stems were bucked under the control of the base price matrix, the shift from bucking-to-value to bucking-to-order optimization with a ±10% price change allowance dropped the sawlog volume by a maximum of 1.4%. Similarly, when the bucking simulator applied bucking-to-value optimization, a volume reduction of 0.9 to 2.3% was produced by replacing Koskitukki’s base price matrix by the price matrix controlled by the real stem data.

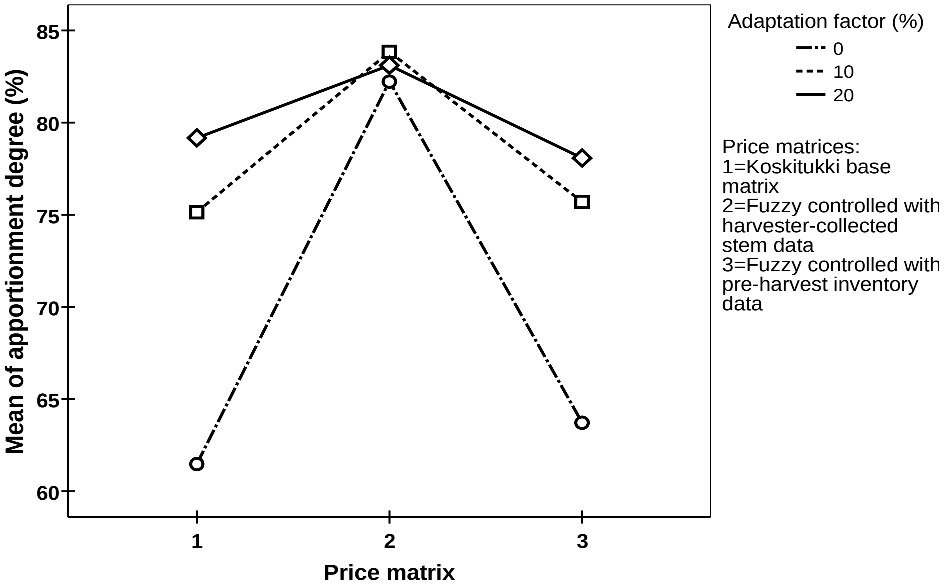

Figure 8. The mean performance of the three price matrices at three adaptation factor levels (0% = bucking-to-value optimization) in four mature Norway spruce (*Picea abies* (L.) Karst.) stands.

Table 4. Analysis of variance of the effects of price matrix and adaptation factor on the apportionment degree.

Source	Sum of Squares	d.f.	Mean Square	F	Sig.
Blocks (stands)	536.279	3	178.760	24.961	<0.001
Main effects:					
Price matrix	942.726	2	471.363	65.818	<0.001
Adaptation factor	825.620	2	412.810	57.642	<0.001
Interaction:					
Price matrix x Adaptation factor	340.010	4	85.002	11.869	<0.001
Error	171.878	24	7.162		
Total	2816.513	35			

4.2 Control of price matrices at forest level (Study II)

Although representing a stochastic process, the GA-based search system behaved quite similarly and thus gave quite similar results in the 10 test runs under the same parameter set (number of iteration cycles = 500, population size = 30, degree of elitism = 1/3, and mutation rate = 0.01). As is usual with evolutionary algorithms, the largest improvements in the solution quality of the price matrix strings generated by the GA system appeared within the first 100 iterations after which the best matrix string at each iteration cycle showed only a small rise in the apportionment degree. The best apportionment degree achieved after 500 iteration cycles varied between 74.1 and 75.5% with a mean of 74.8% and standard deviation of 0.398%. The mean rise in the best apportionment degree on an absolute scale was 15.4 percentage points, from 59.4 to 74.8%

The parameter test exploring the effects of 27 different parameter combinations on the performance of the GA system suggested that a large population size, a low mutation rate and a small amount of elitism be used. Accordingly, the subsequent runs with the GA system for determining stand-specific price matrices were carried out by setting the population size (i.e., the number of price matrix strings in each generation) at 100, the elitism at 33.3% (1/3), and the mutation probability at 0.001 (0.1%).

Strategy 1 performed best at the stand level, as expected. The performance of Strategy 2, however, was not much behind. When the prior control of price matrices was done with the harvester-collected stem data, Strategy 1 outperformed Strategy 2 by 3.9 percentage points on average. When the prior control of price matrices was done using the estimated stem data, the average performance difference between Strategies 1 and 2 was 5.7 percentage points. Strategy 3 performed worst; this is quite natural because in Strategy 3 the log prices were not allowed to change to reflect the differences between the log demand and actual log output distributions observed during the simulation process.

At the forest level, provided that the price matrices were adjusted by the real stem data, all three strategies performed equally well, producing an overall fit of approximately 79%. However, when adjusted by the pre-harvest inventory data, the stand-specific price matrices performed somewhat worse than the uniform price matrix. The reduction in the apportionment degree was 1.1 percentage points when applying Strategy 2 instead of Strategy 1 and 3.6 percentage points when bucking trees under Strategy 3, instead of Strategy 1..

Strategy 2 produced the largest volume of spruce sawlogs at the forest level regardless of whether the price matrix adjustment was done using the real or estimated stem data, with the total spruce sawlog volumes being 5 183 and 5 200 m^3, respectively. Strategy 1 yielded the second largest log volume, 5 166 m^3. The lowest log volumes were given by Strategy 3, being 5 157 m^3 for the price matrices adjusted by the real stem data and 5 159 m^3 for the price matrices using the estimated stem data.[2]

4.3 Control of demand matrices at forest level (Study III)

The stochasticity test showed that the GA system for generating stand-specific log demand distributions behaved much the same as the GA system for generating stand-specific price matrices in Study II. First, the highest overall goodness-of-fit value (GOF_{tot}) found at each

[2] This is wrongly stated in Study II, line 28, 2nd column on page 706.

iteration cycle (i.e., the quality of the best demand matrix string found at each iteration cycle) evolved rapidly within the first 100 iteration cycles (generations) after which only small improvements were seen in the $\mathrm{GOF}_{\mathrm{tot}}$ value. Second, the 10 test runs of the stochasticity test showed no great variation in the maximum $\mathrm{GOF}_{\mathrm{tot}}$ value achieved during the 500 iteration cycles, the maximum $\mathrm{GOF}_{\mathrm{tot}}$ value ranging from 0.189 to 0.201 (from 0.741 to 0.755 in Study II) with a variation coefficient of 0.017 (0.005 in Study II). There was, however, quite a large variation in the number of iteration cycles needed to evolve a matrix string producing the highest goodness-of-fit value at the forest level. Such a matrix string was found at iteration cycle 113 at the earliest and at iteration cycle 492 at the latest. The average number of cycles required was 289.

In the performance test, the overall fit between the log demand and log output distributions at the forest level was higher when the bucking simulations were carried out using the stand-specific demand matrices rather than the uncontrolled reference demand matrices. This was true regardless of whether the stand-specific demand matrices were created using the real or the estimated stem data. When generated by the GA system using the real stem data, the stand-specific demand matrices performed 32% better for the 5%/10 parameter set and 103% better for the 20%/20 parameter set. The corresponding figures for the stand-specific matrices adjusted by the estimated stem data were 22 and 79%. The GA-controlled demand matrices, whether they were generated using the real or estimated stem data, also outperformed the reference demand matrices in most stands for both parameter settings. It was no great surprise that the bucking-to-value optimization performed worst at both the stand and forest level, obviously because no adjustment of log prices to accommodate the log output distributions to the desired ones was allowed while harvesting.

Cutting spruce trees using the bucking-to-order procedure, regardless of whether the bucking process was controlled by the stand-specific or non-stand-specific demand matrices, did not affect the total log volume at the forest level. However, it did affect how the total log volume was split up between different log products. In general, when more ‘freedom’ was given in the bucking-to-order procedure in terms of the maximum deviation and the number of alternative bucking patterns to be explored, there was a larger reduction in the total volume of spruce sawlogs and veneer logs produced (and correspondingly a larger volume of spruce pulpwood logs). When the bucking was done under the control of the 5%/10 parameter set, the volumes of both spruce sawlogs and veneer logs dropped less than 1% from the amounts produced by the pure bucking-to-value procedure applying the uncontrolled reference demand matrices. However, when the 20%/20 parameter was used to control the bucking-to-order procedure there was a moderate rise in the volumes of sawlogs (+18.0 to +22.4%) and pulpwood logs (+22.8 to +34.2%) but a very large drop (-58.1 to -44.9%) in the volume of veneer logs.

4.4 Analysis of four goodness-of-fit measures (Study IV)

There was quite a large variation in the values returned by the four measures for the fit between the same two log distributions. However, one clear regularity was observed. Regardless of the stand and demand matrices used in fitness calculations, the χ^2 measure always provided the lowest fitness value (a range from 0.13 to 0.36) while Laspeyres’ quantity index always produced the highest (a range from 0.67 to 1.23), with the values of the traditional and price-weighted apportionment degrees lying in between these two. For example, when stand C1 (a stand with 380 spruce stems distributed uniformly across all

DBH classes) was cut under the control of demand matrix T1, the similarity between the resulting output distribution and demand distribution T1 was assigned the following values: 1.23 by Laspeyres' quantity index, 0.74 by the traditional apportionment degree, 0.80 by the price-weighted apportionment degree, and 0.33 by the χ^2 measure.

Rank-ordering the 10 generic study stands for both demand matrices according to the fitness scores resulted in the lists shown in Table 5. As can be seen, the only thing all four measures agreed on was that stand A1 (a spruce stand with a normal DBH distribution, a small mean DBH and 380 spruce stems in total) provides the poorest and stand A2 (like stand A1 except that the number of stems was double that of A1) the second poorest fit between the log demand and log output distributions for both demand matrices T1 and T2. Both Laspeyres' quantity index and the price-weighted apportionment degree concluded that stands C1 and C2 (like stand C1 but 760 stems in total) qualify best for the T1 and T2 demand matrices in this particular order. This conclusion was not applicable to the apportionment degree or the χ^2 measure. The apportionment degree indicates that stand B1 (a stand with 380 spruce stems in total, a large mean DBH, and a normal DBH distribution) and E1 (a left-skewed DBH distribution with 380 stems in total) best satisfy the needs of demand matrices T1 and T2 respectively, while the χ^2 measure considered stand B1 as the best choice for both demand matrices.

The behavior of each of the four goodness-of-fit measures was further analyzed by using the fitness values calculated as the decision criterion by which of the two potential demand matrices each stand should be cut (i.e., to which of the two potential sawmills each stand should be directed). Allocating each stand to the alternative providing the highest fitness value yielded the allocation decisions shown in Table 6. All measures agreed that stands A1, B1, B2 (like B1, but twice as many trees), D1 (a right-skewed DBH distribution with 380 stems in total), and E1 should be allocated to sawmill 2 (i.e., cut according to demand matrix T2) and stand C1 to sawmill 1. The allocation strings generated by the traditional and price-weighted apportionment degrees were actually identical and thus also resulted in the same total log value (€290 456). The highest total log value (€290 788) resulted from allocating stands according the χ^2 measure while the lowest value (€290 442) was provided by the allocation based on Laspeyres' quantity index.

The theoretical part of Study IV listed four requirements for an ideal measure for comparing the actual log output distributions to the corresponding demand distributions: (1) comparability of the goodness-of-fit values between stands of different sizes, (2) comparability of the goodness-of-fit values based on the demand matrices of different sizes, (3) aggregation of the product-wise goodness-of-fit values into one stand-wise fitness score, and (4) simplicity and ease of use. Requirements (1) and (2) address the problems often encountered by forest managers in practice of which stand or group of stands suits the given demand matrix best, or vice versa, which demand matrix suits the given stand or group of stands best (i.e., to which mill each stand should be allocated from among several possible choices). Requirement (3) emphasized the ability of a measure to evaluate the goodness of the bucking outcome as a whole, not only for each log product separately. There is no need to mention that an ideal fitness measure should be easy to use and its results should be easy to interpret.

Table 5. The performance order of the 10 generic Norway spruce (*Picea abies* (L.) Karst.) stands (A1, A2,…,E2) for log demand distributions T1 and T2 according to four goodness-of-fit measures. The stands are listed in decreasing order of goodness-of-fit value.

Goodness-of-fit measure							
Apportionment degree		χ^2 measure		Laspeyres' quantity index		Price-weighted apportionment degree	
Demand matrix							
T1	T2	T1	T2	T1	T2	T1	T2
B2	E1	B1	B1	C1	C1	C1	C1
E1	B2	C1	B2	C2	C2	C2	C2
E2	B1	E1	E1	E1	E1	B2	E1
B1	C1	B2	C1	E2	B1	E1	B1
C1	E2	C2	C2	B1	E2	B1	B2
C2	C2	E2	E2	B2	B2	E2	E2
D2	D2	D1	D1	D2	D2	D2	D2
D1	D1	D2	D2	D1	D1	D1	D1
A2	A2	A2	A2	A2	A2	A2	A2
A1	A1	A1	A1	A1	A1	A1	A1

The theoretical analysis concluded that all four measures fully satisfy requirement (3), requirements (1) and (4) partly, and requirement (2) poorly. Basically, because of operating with relative proportions rather than with the actual numbers of logs, each of the four measures takes the stand size into account at least to some extent. Still, two stands with the same number of logs harvested can be quite different in regard to the total number of merchantable trees and can thus perform quite differently in matching the desired log output distribution(s). On the other hand, is this not actually what the fitness measures are originally designed to show? Stands with a small number of stems or a large number of small-sized stems are likely to match the log demand distribution more poorly than stands with a large number of stems and/or a wide DBH distribution. In theory (Koskela et al. 2007), large-size demand matrices (i.e., matrices with a large number of log classes) tend to be much more difficult to satisfy than small-size ones. None of the four goodness-of-fit measures tested, however, can take this fact into account directly. The χ^2 statistic could do this indirectly through the calculation of the statistical significance level (i.e., computing the p-value for the fit between the demand and output distributions). However, the mismatch between the log demand and log output distributions need not be large to cause the χ^2 statistic to judge the distributions entirely different (i.e., rejecting the null hypothesis that the distributions are equal). From the practical point of view, using the p-value as the goodness-of-fit measure is thus not a very good choice. All four fitness scores are relatively easy to compute but some interpretation difficulties may arise because of the large variation in the magnitude of the fitness values between different measures.

Table 6. The decisions made by the four goodness-of-fit measures according to which of the two demand distributions, T1 or T2, each of the 10 generic Norway spruce (*Picea abies* (L.) Karst.) stands should be cut (i.e., to which of the two potential sawmills — sawmill 1 or sawmill 2 — each stand should be directed)

Stand	Goodness-of-fit measure				T1/T2
	Apportionment degree	χ^2 measure	Laspeyres' quantity index	Price-weighted apportionment degree	
A1	T2	T2	T2	T2	0/4
A2	T1	T2	T1	T1	3/1
B1	T2	T2	T2	T2	0/4
B2	T2	T2	T2	T2	0/4
C1	T1	T1	T1	T1	4/0
C2	T1	T2	T1	T1	3/1
D1	T2	T2	T2	T2	0/4
D2	T1	T1	T2	T1	3/1
E1	T2	T2	T2	T2	0/4
E2	T1	T2	T1	T1	3/1
T1/T2	5/5	2/8	4/6	5/5	16/24

5 DISCUSSION AND CONCLUSIONS

5.1 Need for stand-specific demand and price matrices

The primary objective of this study was to test whether the physical fit between the log distributions required by mills and the actual log output distributions from harvesters could be improved by cutting each stand under the control of stand-specific price and demand matrices rather than non-localized reference matrices. The hypothesis was that the price and demand matrices adjusted to the unique conditions of each individual stand would perform better than the uncontrolled reference matrices. This presumption was contrary to the common view that no pre-control of price and demand matrices is needed because of the efficient on-line bucking control systems of modern CTL harvesters.

The findings of Study I and III are contradictory to those of Study II. Study I and III, unlike Study II, gave substantial support to the hypothesis that the stand-specific price and demand matrices would outperform the non-stand-specific matrices in accommodating the log output distribution(s) to the desired one(s). The contradiction between the results of Studies I and II can be mainly attributed to the complementary effect; that is, the log output distributions from several stands are likely to complement each other at the forest level, with the cumulative output distribution thus providing a better match to the demand matrix than the output distribution from only one stand. All three studies, however, agree that improving the physical fit between the log demand and log output distributions requires relaxing the aim of producing the maximum output volume for each log product.

Given the findings of Studies I-III, we may then conclude that

1. At the stand level, the pre-control of price matrices seems to be advantageous, provided that the stem profile of each tree in a stand is known or can be estimated reliably.
2. When the comparison between the log demand and log output distributions is made at the forest level, the stand-specific and non-stand-specific price matrices seem to perform equally well.
3. Some gain seem achievable at the forest level by dividing the overall demand matrices into stand-specific sub-demands and applying the bucking-to-order method while harvesting.

An important question is how valid and conclusive are the above-stated conclusions? Answering this question requires carefully considering at least the following issues related to the materials and methods employed in Studies I-III. (1) Were the control systems used for generating stand-specific price and demand matrices efficient enough; did they find the most optimal matrix or matrix combination for each stand involved in the optimization process? (2) Were both the reference and stand-specific price and demand matrices/matrix combinations exposed to sufficiently comprehensive and severe tests? Specifically, were the stands and the overall log demand distributions, on which the performance comparison between the various matrix sets was based, representative enough to allow the generalization of the study findings? (3) Are the results from the bucking simulations reliable; did the changes observed in the fitness values between the log demand and log output distributions really result from the changes in either the price or demand matrices or is there any possibility that the fitness changes have been caused by some uncontrolled factors? (4) How was the bucking-to-order control method implemented in the bucking simulators used for testing the performance of the different price/demand matrix settings; were the algorithms employed by the simulators efficient enough?

Applying modern heuristics such as fuzzy logic and genetic algorithms to solving practical or scientific problems requires deciding on the values of numerous parameters controlling the problem-solving process. In a fuzzy control system, for example, one has to decide on the number, shape and locations of the membership functions for both input and output variables, the number and types of fuzzy inference rules (if-then rules), how to fuzzify the crisp input values, how to perform the fuzzy reasoning for each fuzzy rule given the fuzzified input values (whether to use Sugeno or Mamdani type reasoning and how to interpret different logical operations), how to aggregate the outputs of individual fuzzy rules and, finally, how to defuzzify the aggregate fuzzy output value. Similarly, when working with genetic algorithms, one will face problems such as how to encode the candidate solutions, how to evaluate their appropriateness to solving a given problem (an evaluation function), which variation operators (crossover, mutation, inversion) to use and how to implement them, when to stop the GA process (a stopping criterion), and how large to set the population size.

Unfortunately, there is no universally optimal parameter set for both fuzzy control and GA systems that would work fine for all kinds of problems, nor a universally valid theory that would guide one through the process of finding such a parameter set (see Puolakka

1997, Michalewicz and Fogel 2000). That is to say, each unique problem usually requires a unique parameterization which can be found only by experimenting with different parameter value combinations. As there are many parameters involved in fuzzy and GA systems, each of them normally having a wide range of possible values, experimenting on the performance of all possible parameter settings is certainly a huge, if not impossible task, even with today's fastest supercomputer. Rather than trying to figure out the best possible parameter setting before starting to run the problem-solving system, we may instead configure the system or design it to configure itself, at least partially, while still seeking the solution to the original problem. Online adaptation or self-adaptation of parameters seems to be an effective approach especially for GA applications because they intrinsically represent adaptive, dynamic processes (Michalewicz and Fogel 2000). This online parameter control mechanism was, however, not embedded in the GA systems of Studies II and III.

Considering all this, it is quite certain that both the fuzzy control system of Study I and the GA search systems of Studies II and III were at least to some extent inefficient and non-optimal. Knowing exactly the degree of non-optimality of the solutions suggested by the control systems of Studies I-III would have required that all possible log output distributions for each product in each stand had been enumerated. Although all three systems were constructed by exploring their performance under different parameter settings, the number of different settings tested was quite limited. Thus, it is quite possible that, although based on commonly applied settings, the parameter settings selected for the performance tests in Studies I-III might have been far from the most optimal and efficient ones.

The largest search for the 'right' parameter configuration was carried out in Study II. This search included 27 different parameter settings (3 parameters at 3 levels). In Study III, on the other hand, no parameter testing was included, the parameter values being set at those commonly applied in the GA community. In Study I, after the first trials the system construction focused on fine-tuning the same two-input one-output fuzzy inference system, representing five fuzzy sets (linguistic states) for all three variables. Changing the locations and shapes of the membership functions as well as planning the rule base were, however, done subjectively. For example, in the case of conventional fuzzy process control systems, the normal tuning approach relies on systematically analyzing the control signal(s) and the deviation(s) between the set-point value(s) and the actual output value(s) as a function of the control iterations (Puolakka 1997). Although able to show what is wrong with the fuzzy system, an analysis of this kind cannot explicitly show how to change the configuration of the system to make it perform in a desirable way; large experiments with different set-ups might still be necessary.

Clearly, Studies I to IV each represent a case study. First, only one overall log demand distribution per each log product was included in the performance tests in each study. While these overall demand matrices came from real sawmills and veneer mills and can thus be considered quite representative, they were certainly not shared by all such Finnish plants operating at the time of the data collection. As each sawmill normally defines its own demand matrices on the basis of its production strategy, the current market situation and the characteristics of timber available at its main procurement area, there may be marked differences in the size and/or contents of the demand matrices between the sawmills included in the study and those not included. Second, the 15 mature Norway spruce study stands shared by Studies I-III were all located within a geographically restricted area in southern Finland. They were probably good representatives of the mature spruce stands in that particular area but not necessarily of those in other parts of Finland. That is to say, the results from the bucking tests in Studies I-III could have been totally different if different stands and/or overall demand matrices had been used. Making generalizations of any kind from the results obtained in Studies I-III is thus quite questionable and is not recommended due to the small number of stands and log products included in the studies.

Making far-reaching generalizations from the results of Studies I-III are also prevented by the fact that the results were obtained in an ideal ‘laboratory’ rather than real forest circumstances. In actual operating conditions trees are seldom fault-free, harvesters can seldom predict and measure the shape of a tree perfectly, and the harvester and mill measurement systems do not necessarily assign the same dimensional attributes to a given log. Excluding these facts, however, guarantees that the differences observed in the goodness-of-fit values were really caused by the differences in price-demand-matrix settings, and not, for example, by different harvesting conditions, optimization algorithms or harvester operators. An important and justifiable question then comes: how would have the stand-specific price/demand matrices performed under real harvesting conditions in comparison to the non-localized reference matrices? No doubt, both matrix types would likely have produced a poorer fit between the log demand and log output distributions in real harvesting situations than they did in the theoretical bucking simulations. This is mainly for two reasons. (1) The errors occurring in stem length and diameter measurements and/or model errors in stem shape predictions make the harvester select non-optimal bucking patterns. Vuorenpää et al. (1997), for example, reported that the apportionment degree at the stand level dropped by at most 5% when the bucking of trees was based on predicted rather than measured stem profiles. (2) The bucking patterns suggested by the harvester’s bucking computer cannot always be implemented because of poor quality stems. In a real harvesting situation we usually do not have perfect knowledge of the stand composition available and are thus obliged to perform the control of price/demand matrices using relatively unreliable estimates of stand structure. As shown in Studies I-III, the matrices adjusted by imperfect stem data seldom performed much better than the uncontrolled reference matrices.

It is not known how exactly the bucking-to-order method in the Ponsse OptiSimu bucking simulators works. According to the Ponsse company, the bucking algorithm itself follows the approach taken by Näsberg (1985), while the bucking-to-order optimization is implemented through the adaptive price list technology. Näsberg’s algorithm, however, is a pure bucking-to-value algorithm and cannot thus give any hint on how to implement the adaptive price list approach. Also, the Ponsse company has not revealed how they have coded the on-line price matrix adjustment process. It is thus quite impossible to evaluate how efficiently the Ponsse OptiSimu simulators performed the price matrix adjustment. Thus, all that can be said is that the Ponsse OptiSimu version 2.50, which was used in Study I, was probably not as efficient in adjusting the price matrices as the other two OptiSimu versions used in Studies II and III. This suspicion is reasonable because version 2.50 was among the very first simulator models ever designed by the Ponsse group and can thus be regarded at least to some extent as a prototype. This means that the apportionment degrees obtained with the bucking-to-order optimization would probably have been somewhat higher for both fuzzy controlled and reference price matrices if some more advanced simulator version had been available at the time of the study. The algorithm of the VP-Simu bucking simulator and the bucking modules embedded in the GA search systems in Studies II and III were verified by comparing the bucking-to-value results produced by these algorithms to those obtained with the Ponsse OptiSimu simulator for the same price matrix set. This analysis showed only slight differences between the resulting output distributions, probably caused by some rounding differences occurring during the valuation of different bucking patterns.

Considering the shortcomings related to the quality of the control systems developed, and the amount and quality of the data as well as the methods and tools these systems were tested on, the conclusions made concerning the usefulness of drawing stand-specific price and demand matrices must be regarded only as preliminary rather than conclusive.

As stated in the Introduction, few studies have addressed the issue of whether or not the bucking work on harvesters should be controlled by customized rather than non-customized price and/or demand matrices. In their early study, Vuorenpää et al. (1997) compared the ability of six different price matrices to produce two different Norway spruce sawlog distributions in three different spruce stands. The price matrices tested were not especially stand-oriented, but merely offered alternative overall price matrices. In the same work, Vuorenpää et al. (1997) also examined whether the bucking outcome as a whole could be improved by classifying the stands to be harvested into a few stand types and by assigning each stand type its own demand matrix. The stand-type specific demand matrices were generated by keeping the log length distribution within each SED class similar to that of the overall demand matrix while allowing the target log proportions in each SED class to vary according to the stand type. In their later study, Vuorenpää et al. (1999) compared the performance of four different price matrices in producing the desired output log mix for Norway spruce sawlogs in one thinning stand and two mature stands.

The results and conclusions of Vuorenpää et al. (1997, 1999) partly parallel, partly contradict those of this study. The bucking simulations conducted in the early study of Vuorenpää et al. indicated that, with few exceptions, the customized price matrices outperformed the uniform price matrix in all stands for both demand matrices. The simulation results of their later study, however, showed no large differences in the ability of the tested price matrices to produce the desired output matrix, the stand-level apportionment degree values varying between 0.91 (91%) and 0.93 (93%). The overall conclusion of Vuorenpää et al. was that all stands can be cut under the control of the same price matrix; no adjustment of price matrices is needed. Although the stand-type specific demand matrices showed some improvement in performance over the non-specific demand matrix shared by all stands, Vuorenpää et al. (1997) concluded that there is no need to divide the overall demand distribution into stand or stand-type specific sub-demand distributions. An identical conclusion was drawn in their later study.

5.2 Methods for generating stand-specific demand and price matrices

At first glance, looking at the titles of Studies I-III, a reader of this thesis may easily get an impression that this study is entirely about fuzzy logic and genetic algorithms. As has been seen above, this impression is not correct. However, due to the large number of different techniques and heuristics available for problem solving the following questions require answers: (1) why should we use fuzzy logic or genetic/evolutionary algorithms in generating stand-specific bucking instructions; and (2) would the other search and optimization techniques available have produced better-performing price and demand matrices?

There are no sound reasons for not using problem-solving methods other than fuzzy logic and GAs. Fuzzy logic was selected as the control technique in Study I mainly for two reasons. First, fuzzy logic in general had been reported to have performed successfully in solving a wide variety of control problems in many problem areas (Niskanen 1993, Puolakka 1997). Second, because there was no clear idea of how exactly log prices should be adjusted during the calibration process of a price matrix (except that the prices of the log classes showing a surplus of material should be lowered and vice versa), it was assumed that, when built on a fuzzy approach, a workable control system might be readily available. Constructing a workable conventional control system using a stair-wise adjustment policy, for example, might have been a much more complex task. This is because for a conventional control system to perform as well as a fuzzy control system, a larger number of control rules and/or more complex ones is/are required (Niskanen 1993). On the other hand, the task of finding an optimal price matrix, even for one log product in only one stand, presents a highly combinatorial optimization problem. For example, given that there are 10 possible values to be assigned to each diameter-length class in a price matrix comprising 200 log classes, the size of the search space would be as large as 10^{200}. Many combinatorial problems of a similar size and type have been solved through a genetic algorithm approach (e.g., Alander 1998).

No doubt, fuzzy logic and genetic algorithms present only two possible methods of accommodating the price and/or demand matrices to the unique structure and tree characteristics of each stand. The other techniques that could have been used include both conventional approaches, such as the two-stage LP/DP method (or the LP/SP method as Laroze (1999) calls it), and modern soft computing tools, such as simulated annealing, tabu search and neural networks. These methods, as well as fuzzy logic and genetic algorithms, could also have been combined together to form hybrid systems such as a fuzzy-genetic, neural-fuzzy or genetic-neural approach. A hybrid neural-fuzzy system, for example, is a system in which the fuzzy logic unit responsible for accomplishing the actual problem solving task is tuned using a neural network.

Wolpert and Macready (1995) (see also Macready and Wolpert (1995), and Alander (1998)) in their theoretical study have shown that, on average, there is no algorithm that will always outperform other algorithms for any given search problem. This finding is known as the No Free Lunch (NFL) Theorem. Thus, while GAs, for example, have widely been reported to have performed well in solving combinatorial optimization problems of many types, a genetic algorithm approach may not necessarily be the best choice in seeking optimal stand-specific price and demand matrices at the forest level. That is, some or all of the alternative search techniques listed above might have produced better price and demand matrices. The problem, however, is that there is in general no way to know the problem-solving performance of each particular algorithm for any given problem accurately in advance. The problems that can be solved using linear programming form an exception to this rule because we can always count on the simplex algorithm returning the optimal solution. As demonstrated by Näsberg (1985), the linear programming technique could have also been applied, at least at the stand level, to generating stand-specific price matrices. The main problem with the LP/DP technique lies in the difficult implementation of its solutions. An LP/DP solution typically lists several price matrices (= bucking patterns) for each stem class, which constitutes a new decision problem: which price matrix should be applied to bucking a tree in a given stem class. The size of the resulting LP/DP models, at least at the forest level, would certainly also have been enormous.

5.3 Measuring the fit between the log demand and log output distributions

Just as there are different techniques available for price and demand matrix optimization, there are likewise several measures available for evaluating the fit between the log demand and log output distributions. And just as there seems to be no superior problem-solving algorithm, there likewise seems to be no superior metrics for measuring the similarity between the log demand and log output distributions.

It should, however, be noted that the test results of Study IV only showed that all four fitness measure candidates (1) met three of the four important requirements listed for the ideal fitness measure, at least partly (all measures seem to lack the ability to 'normalize' the fitness values according to the demand matrix sizes), and (2) provided quite consistent results for different demand matrices in different stand types. These results thus do not automatically imply that it makes no difference which measure one applies to assess the fit between the log demand and log output distributions.

To address this issue properly, we would need a measure of measures and/or test arrangements enabling comparison between the performances of different fitness measures.

An attempt of this kind was conducted by Malinen and Palander (2004), who compared how well the bucking-to-order procedure under the control of five different similarity measures succeeded in meeting the desired log output at the stand level. The idea in their close-to-optimal bucking algorithm was that, rather than the overall priority indices, the stem-level selection between bucking patterns is done according to the overall fit between the demand distribution and the cumulative output distribution resulting from each bucking pattern suggested. The five goodness-of-fit measures tested for controlling the close-to-optimal bucking procedure were: (1) a conventional apportionment degree (Malinen and Palander called this measure a distribution level, DL for short); (2) a penalty segmented version of the previous measure (called PSDL); (3) a squared apportionment degree (or a squared distribution level, SDL); (4) a standard chi-square (χ^2) statistic (the same as used in Study IV); and (5) a flexible penalty segmented apportionment degree (called FPSDL). The stand-level fit between the desired log output distribution and the final cumulative output distribution was evaluated using both the conventional (DL) and squared apportionment degrees (SDL).

The best matches between the log demand and log output distributions at the stand level were obtained by applying either the squared apportionment degree, the χ^2 statistic, or the flexible penalty segmented apportionment degree (FPSDL) as a decision criterion for the stem-level bucking pattern selection.

As was the case in Study IV, however, there was no great variation in the stand-level goodness-of-fit values between the measures tested, with the DL values varying between 86.1% and 87.8% and the SDL values between 96.3% and 99.3%. In this respect, Malinen and Palander's results are in quite a good accordance with the findings of Study IV.

While each of the goodness-of-fit measures suggested so far for assessing the fit between the log demand and log output distributions seem to be suitable for use in actual wood procurement, one should keep the following things in mind when considering appropriate fitness measures and evaluating results.

First, each measure has its advantages and disadvantages. A conventional apportionment degree, for example, is easy to use and interpret while it considers all SED-length classes of logs as equally important. The χ^2 measure, on the other hand, does not suffer from a similar problem but, because based on a statistical test, this measure seems to be very sensitive to the differences between the actual and desired log frequencies. The index theory based approach (Laspeyres' quantity index) and the price-weighted apportionment degree differ from the two above in that they offer an opportunity of incorporating the economic aspect into the fitness assessment procedure. What this means is that the selection between appropriate fitness measures is to a large extent a matter of both user preferences and reasons for recording and monitoring the fit between the log demand and log output distributions

Second, when comparing the goodness-of-fit values achieved by harvester models operating in various areas and (saw)mills, it should be kept in mind that cutting two different stands under the control of different demand matrices actually comprises two quite distinct actions. It follows from this that the resulting goodness-of-fit values are seldom fully commensurable and should thus not be applied blindly without careful analysis.

Third, the log output distribution at both stand and forest levels seldom matches fully the demand distribution. The fitness scores returned by the currently available fitness measures reflect how far the output remains from the demand. They do not, however, tell anything about the economic effects resulting from oversupplying or undersupplying the log class entries in the demand matrices.

5.4 Final remarks and future perspectives

In any optimization, the quality of the input data is of ultimate importance. An old aphorism in computer science puts it this way: ”Garbage in, garbage out”. This is exactly what is likely to happen with the FLC and GA systems of Studies I-III if initiated with imperfect, biased stem data. Thus, designing and implementing systems for the prior control of price and/or demand matrices make sense only if sufficiently reliable tree-level information on forest stands is available in advance.

Prior knowledge on forest stands can be acquired in many ways. The traditional approach, which was also used in this study, is to make a special stand inventory, either separately or in conjunction with the harvest or forest management planning activities. Because it is time-consuming and hence expensive, this traditional data acquisition method has largely lost its early dominance. The other way to produce stem-level stand information is to apply indirect computational methods.

In Finland, for example, forest management plans cover approximately two-thirds of the private forest land area (Karppinen et al. 2002). A tree population for a given stand can then be compiled from the management plan through theoretical stand- and tree-level models. Another computational approach which has attracted much interest in Finland in recent years (Malinen et al. 2001, Malinen 2003, Räsänen et al. 2000, Räsänen et al. 2005) is to make use of the stem data measured and stored by harvesters during their every-day cutting work. That is, given some prior information on the structure and tree characteristics of a stand scheduled for harvesting, an attempt can be made to find one or more stands from a database of previously cut stands which, in terms of the search variables (e.g., species mixture, basal area by tree species, stand area and age of trees) resemble the stand in question. If found, these most similar neighbor stands, either as such or after some further processing, are then used as an estimate for the uncut stand.

All the above estimation methods assume that someone has visited the stand in the immediate past. Tree-level prior information on stands can, however, be gathered without making any such trips by means of modern remote sensing techniques. This is exemplified, for example, by Korpela (2004) whose tree-level forest inventory approach employs multiple digitized aerial photographs and advanced image interpretation algorithms for positioning tree tops, recognizing tree species and measuring the height and crown width of trees.

Whatever system we may use for estimating the structure and tree characteristics of stands to be harvested, the estimates seldom match the real stand conditions perfectly, at least at the tree level (see Korpela 2004, Räsänen et al. 2005). It also seems quite apparent that no breakthrough will be seen in this field in the near future. Thus, rather than trying to generate optimal bucking instructions for every stand in a harvesting plan, it might be better to divide the stand population into a few sub-populations (stand-type groups) and associate each of these with group-specific demand and/or price matrices. In an approach of this kind, the unavoidable inaccuracy of the prior information might not lead to such severe sub-optimization as it usually does when generating stand-specific bucking matrices.

Even if there were no differences at all between the estimated and real stem data, and thus the stand or group-specific bucking instructions assigned to harvesters were fully optimal, the actual log output distributions from harvesters might still be far from the mills' demand matrices. This is, as stated earlier, mainly for two reasons. First, random and/or systematic errors in both measuring and predicting the profiles of tree stems cannot be entirely avoided. Second, when harvesting stands, one can hardly avoid encountering stem defects of various kinds. As a result, the harvester's bucking system often suggests, or alternatively, the harvester operator is often forced to make suboptimal bucking decisions. Both situations effectively prevent achieving the desired output distributions. What is needed is (1) better measurement systems for harvesters, (2) better systems for stem profile prediction (note that the performance of prediction models is affected not only by the model itself but also by the accuracy of the stem measurements for the first 3 to 4 metres from the butt), and (3) better prior information systems, offering data not only on the dimensional but the qualitative features of trees in stands. It should also be worth testing whether the present on-line control systems on harvesters (i.e., the adaptive price list approach and the close-to-optimal method) are the best choices for accommodating the log output distributions to the desired log distributions. For example, because the reasoning about the appropriate on-line control actions is made under uncertainty (the harvester's information system does not know the properties of the trees to be logged next), a fuzzy logic system might be one potential alternative to implement the on-line control of the bucking process on harvesters.

This study addressed only a tiny link in the whole logistic chain from forest to mill. A perfect match between the log output distributions and the demand distributions does not automatically imply that the whole timber supply chain from forest to mill will work optimally, especially in terms of cost. This is because the maximum fit between the log demand and actual log output distributions can, in most cases, be achieved only by relaxing the primary aim of minimizing harvesting and transportation costs (Imponen 1999). The obvious question then is whether the better fit between the log demand and log output distributions results in increased profits, thus compensating for an increase in timber supply cost. Thus, to thoroughly optimize the whole production chain from forest to mill and even to end customer, a holistic model is needed that would consider (1) which products to cut in each stand available for harvesting (the product allocation between stands), (2) what diameter, length and quality of logs to cut for each product in each stand (the log allocation between stand), and (3) in what order to cut the stands, with the overall aim being to maximize the difference between the revenues and costs.

Many questions in the field of bucking optimization are thus still open – awaiting answers and eager researchers.

LITERATURE CITED

Ahonen, O.-P. & Lemmetty, J. 1995. Leimikon tukkijakauman ohjauksen keinot. Summary: Means of controlling log distribution within a harvesting site. Metsäteho katsaus 5. 6 p.

Airola, N., Happonen, E., Jorkama, M., Kojo, T., Komulainen, P., Luomi, S., Malinen, U., Paanasalo, J., Rautakorpi, T., Turunen I. & Veräjänkorva, J. 1999. Reeling and winding. In Papermaking Part 3, Finishing. Book 10. Fapet Oy, Helsinki, Finland. pp. 141-236.

Alander J.T. 1998. Geneettisten algoritmien mahdollisuudet. Summary: The potential of genetic algorithms. Teknologiakatsaus 59. Tekes, Helsinki. 102 p.

Anderson, D.R., Sweeney, D.J. & Williams, T.A. 1994. An Introduction to Management Science: Quantitative Approaches to Decision Making. 7th edition. West Publishing Company. St. Paul, Minneapolis. 1002 p.

Arce, J.E., Carnieri, C., Sanquetta, C.R. & Filho, A.F.. 2002. A forest-level bucking optimization system that considers customer's demand and transportation costs. For. Sci. 48(3): 492-503.

Bare, B.B., Briggs, D.G., Roise, J.P. & Schreuder, G.F. 1984. A survey of systems analysis models in forestry and the forest product industries. Eur. J. Operat. Res. 18: 1-18.

Bailey, R.L. & Dell, T.R. 1973. Quantifying diameter distributions with the Weibull function. For. Sci. 19: 97-104.

Beasley, D., D.R. Bull & R.R. Martin. 1993a. An Overview of Genetic Algorithms: Part 1, Fundamentals. University Computing 15(2): 58-69.

—, D.R. Bull & R.R. Martin. 1993b. An Overview of Genetic Algorithms: Part 2, Research Topics. University Computing 15(4): 170-181.

Bergstrand, K.-G. 1989. Fördelningsaptering med näroptimal – Reviderad version [Bucking to order with a close-to-optimal method – Revised version]. Forskningsstiftelsen Skogsarbeten. 1989-12-11. 11 p. [In Swedish.]

— 1990. Fördelsninsaptering – ett sätt att tillgodose sågverksönskemål [Bucking to order – a way to satisfy sawmill demands]. Forskningstiftelsen Skogsarbeten. Resultat 11. [In Swedish.]

Bobrowski, P.M. 1990. Branch-and-Bound Strategies for the Log Bucking Problem. Decision Sciences 21(1): 1-13.

— 1994. The log bucking problem: A comparison of dynamic programming versus branch and bound. Eur. J. Operat. Res. 74: 495-508.

Boston, K. & Murphy, G. 2003. Value Recovery from Two Mechanized Bucking Operations in the Southeastern United States. South. J. Appl. For. 27(4): 259-263.

Bowers, S. 1998. Increased value through optimal bucking. Western Journal of Applied Forestry. 1998; 13(3): 85-89.

Briggs, D.G. 1977. A dynamic programming model for bucking tree stems into logs. Institute of Forest products, Contribution no. 30. University of Washington, Seattle, WA. 12 p.

— 1980. A dynamic programming approach to optimizing stem conversion. Ph.D. dissertation. College of Forest Resources, Univ. of Washington, Seattle, WA. 397 p.

Clemmons, J.W. 1966. New technique in individual tree description for the basis of economic decision making. In Proceedings of the Society of American Foresters' Annual Meeting, Seattle, Washington. The Society of American Foresters, Bethesda, MD. pp. 197-200.

Coggman, I. & Gustafsson, J. 1985. Bilaga: Simuleringsprov med en metod för datorstödd specifikationsaptering [Appendix: Simulation Tests on Computerized Bucking to Specification]. In Aptering med dator – Ett sätt att höja virkesvärdet vid maskinell avverkning [Computerized bucking – One way to Increase the Value of the Wood in Mechanized Logging Systems]. Berglund, H., and J. Sondell. Forskningsstiftelsen Skogsarbeten. Redogörelse 6. pp. 43-51. [In Swedish.]

Coyner, B. 2004. Moving From Cut-to-Length to Cut-to-Diameter [Online journal]. Available from: http://www.forestnet.com/timberwest/archives/May_June_04/moving_from.htm [cited 22 February, 2006].

Dantzig, G.B. & Wolfe, P. 1960. Decomposition Principle for Linear Programs. Operations Res. 8(1): 101-111.

— & Wolfe, P. 1961. The Decomposition Algorithm for Linear Programs. Econometrica 29(4): 767-778.

Eisemann, K. 1957. The Trim Problem. Management Sci. 3(3): 279-284.

Eng, G. & Daellenbach, H.G. 1985. Forest Outturn Optimization by Dantzig-Wolfe Decomposition and Dynamic Programming Column Generation. Operations Res. 33(2): 459-464.

—, Daellenbach, H.G. & Whyte, A.G.D. 1986. Bucking tree-length stems optimally. Can. J. For. Res. 16: 1030-1035.

Faaland, B. & Briggs, D. 1984. Log bucking and lumber manufacturing using dynamic programming. Management Science 30(2): 245-257.

Fobes, E.W. 1960. Quality-Controlled Log Bucking Produces High-Grade Logs and Top Lumber $$$. Forest Prod. J. 10(8): 415-418.

Ford, L.R., Jr. & Fulkerson, D.R. 1958. A Suggested Computation for Maximal Multi-Commodity Network Flows. Management Sci. 5(1): 97-101.

Forster, R.B. & Callahan, J.C. 1968. Allocating Indiana hardwood timber in multiple product situations. Purdue University, Agriculture Experiment Station, Research Bulletin 843. Lafayette, IN. 18 p.

Fuzzy logic toolbox user's Guide. 1998. Ed. 3. The MathWorks, Inc., Natick, MA. 237 p.

Garland, J., Sessions, J. & Olsen E.D. 1989. Manufacturing logs with computer-aided bucking at the stump. Forest Prod. J. 39(3): 62-66.

Gilmore, P.C. & Gomory, R.E. 1961. A Linear Programming Approach to the Cutting-Stock Problem. Operations Res. 9(6): 849-859.

— & Gomory, R.E. 1963. A Linear Programming Approach to the Cutting-Stock Problem – Part II. Operations Res. 11(6): 863-888.

Glück, P. & Koch, W. 1973. Die optimale Rohholzausformung. Summary: Optimal Tree Bucking. Cbl. Fortswesen 90: 193-228.

Godin, A.E. 2001. Logging equipment database: 2001 update. Advantage 2(59). 2 p.

Gobakken, T. 2000. The Effect of Two Different Price Systems on the Value and Cross-cutting Patterns of Norway Spruce Logs. Scand. J. For. Res. 15: 368-377.

Greene, W.D., Jackson, B.D. & Culpepper, J.D. 2001. Georgia's logging businesses, 1987 to 1997. Forest Products Journal 51(1): 25-28.

Grondin, F. 1998. Improvements of the dynamic programming algorithm for tree bucking. Wood and Fiber Science 30(1): 91-104.

Hans, E.W. 2001. Resource Loading by Branch-and-Price Techniques. Ph.D. thesis, Beta Research School for Operations Management and Logistics, University of Twente. Twente University Press, the Netherlands. 164 p.

Holland, J.H. 1992. Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence. MIT Press, Cambridge, MA. 211 p.

Imponen, V. 1987. Pölkytysongelman ratkaisu askelittaisella optimoinnilla. Summary: Solving the problem of bucking through optimizing step by step. Metsätehon katsaus 22. 4 p.

— 1999. Puutavaralogistiikka pelkistää hankinnan toimintamallit [Timber logistics simplifies the operations models of wood procurement]. Metsätieteen aikakauskirja 4: 722-726. [In Finnish.]

— 2001a. Katkonnan ohjauksen periaatteet [Principles of Tree Bucking Control]. In Proceedings of the 1st Conference: Evolving Wood Supply 2001 – Conference for Forest Professionals. Vantaa, Finland. 7-8 February 2001. Edited by J. Hämäläinen, M. Gustafsson, A. Kariniemi, and R. Leustu. Metsäteho, Helsinki. pp. 84-90. [In Finnish.]

— 2001b. Puunhankinnan prosessien tuotelähtöinen uudistaminen – ovatko valmiudet olemassa? [Product-oriented reformation of the wood procurement processes: Do we have the facilities available?] Seminaariesitelmä Wood Wisdom –tutkimusohjelman Puu- ja paperituotteiden arvoketjujen optimointi ja hallinta –seminaarissa. [Presentation at Wood Wisdom Research programme Seminar on the optimization and management of the value chains within the wood and paper product industry.] Wood Wisdom Research Programme, Helsinki. 21 August, 2001. 6 p. [In Finnish.]

Karppinen, H., Hänninen, H. & Ripatti, P. 2002. Suomalainen metsänomistaja 2000 [Finnish forest owner 2000]. Metsäntutkimuslaitoksen tiedonantoja 852. 83 p. [In Finnish.]

Klir, G.J. & Yuan, B. 1995. Fuzzy sets and fuzzy logic: theory and applications. Prentice Hall, New Jersey. 574 p.

Korpela, I. 2004. Individual Tree Measurements by Means of Digital Aerial Photogrammetry. Silva Fennica Monographs 3. 93 p.

Koskela, L., Sinha, B.K. & Nummi, T. 2007. Some aspects of the sampling distribution of the apportionment index and related inference. Manuscript. Submitted to Silva Fennica.

Kärkkäinen, M. 1986. Miten rungot tulisi apteerata. Summary: How to determine sawlog length. Työtehoseuran metsätiedote 15. 4 p.

Laasasenaho, J. 1982. Taper curve and volume functions for pine, spruce and birch. Communicationes instituti forestalis Fenniae 108. The Finnish Forest Research Institute. 74 p.

— 1996. Metsän mittaus ja kartoitus [Forest mensuration and mapping]. Lecture notes for MARV1 course at the department of Forest Resource Management, University of Helsinki, Helsinki. 152 p. [In Finnish.]

Lappi, J. 1991. Calibration of Height and Volume Equations with Random Parameters. For. Sci. 37(3): 781-801.

Laroze, A. 1999. A linear programming, tabu search method for solving forest-level bucking optimization problems. For. Sci. 45(1): 108-116.

— & Greber, B.J. 1997. Using Tabu Search To Generate Stand-Level, Rule-Based Bucking Patterns. For. Sci. 43(2): 157-169.

MacDonald, A.J. 1999. Harvesting Systems and Equipment in British Columbia. FERIC Handbook No. HB-12. B.C. Ministry of Forestry, Victoria, BC, Canada. 211 p.

Macready, W.G. & Wolpert, D.H. 1995. What Makes An Optimization Problem Hard? Working Papers 95-05-046. The Santa Fe Institute, Santa Fe, NM. 11 p.

Malinen, J. 2003. Locally Adaptable Non-parametric Methods for Estimating Stand Characteristics for Wood Procurement Planning. Silva Fennica 37(1): 109-120.

—, Maltamo, M. & Harstela, P. 2001. Application of Most Similar Neighbor Inference for Estimating Marked Stand Characteristics Using Harvester and Inventory Generated Stem Databases. International Journal of Forest Engineering 12(2): 33-41.

Malinen, J. & Palander, T. 2004. Metrics for Distribution Similarity Applied to the Bucking to Demand Procedure. International Journal of Forest Engineering 15(1): 33-40.

Marshall, H.D. 2005. An Investigation of Factors Affecting the Optimal Log Output Distribution from Mechanical Harvesting and Processing Systems. Ph. D. thesis, Oregon State University, Corvallis, OR. 210 p.

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

Michalewicz, Z. 1996. Genetic algorithms + data structures = evolution programs. 3rd ed. Springer-Verlag, Berlin. 387 p.

— & Fogel, D.B. 2000. How to Solve It: Modern Heuristics. Springer-Verlag, Berlin. 480 p.

Mitchell, M. 1996. An introduction to genetic algorithms. The MIT Press, Massachusetts Institute of Technology, Cambridge, Massachusetts. 218 p.

Murphy, G.E. 2002. Mechanization and value recovery: worldwide experiences. In Proceedings of the Woodfor Africa 2002, Forest Engineering Conference: Forest Engineering Solutions for Achieving Sustainable Forest Resource Management - An International Perspective, Hilton College, Pietermaritzburg, South Africa, 2-3 July, 2002. Edited by Loren Kellogg, Ben Spong, and Priscilla Licht, Oregon State University, Department of Forest Engineering, Corvallis, Oregon. pp. 23-32.

—, and Olsen, E.D. 1988. Value recovery from trees bucked on a landing and at the stump. Forest Prod. J. 38(9): 49-52.

—, Marshall, H. & Bolding, M.C. 2004. Adaptive control of bucking on harvesters to meet order book constraints. Forest Prod. J. 54(12): 114-121

Möller, J.J., Sondell, J. & Arlinger, J. 2002. Virkesvärdestest 2001 – Apteringsfrågor. Summary: Timber value trials 2001: Merchandising aspects. SkogForsk Redogörelse 7. 52 p.

Nilsson, N. & Sondell, J. 1987. Dapt 520 apteringssystem för värdeaptering [Dapt 520 bucking system for bucking-to-value optimization]. Forskningsstiftelsen Skogsarbeten. Resultat 22. [In Swedish.]

Niskanen, V.A. 1993. Sumeat järjestelmät [Fuzzy systems]. In: Hyvönen E., Karanta, I. & Syrjänen, M. (eds.). Tekoälyn ensyklopedia [Encyclopedia of Artificial Intelligence]. Gaudeamus, Hämeenlinna. pp. 154-165. [In Finnish.]

Nordmark, U. & Oja, J. 2004. Prediction of Board Values in *Pinus sylvestris* Sawlogs Using X-ray Scanning and Optical Three-dimensional Scanning of Stems. Scand. J. For. Res. 19: 473-480. DOI: 10.1080/02827580410030172

Näsberg, M. 1985. Mathematical programming models for optimal log bucking. Linköping Studies in Science and Technology. Dissertation. No. 132. Department of Mathematics, Linköping University, Linköping. 201 p.

Näslund, M. 1936. Skogsförsöksanstaltens gallringsförsök i tallskog: primärbearbetning [Experiments of the Swedish Institute of Experimental Forestry in harvesting pine stands: preliminary results]. Meddelanden från Statens skogsförsökanstalt 29(1). 169 p. [In Swedish with German summary.]

Olsen, E.D., Pilkerton, S.J., Garland, J. & Sessions, J. 1991. Computer aided bucking on a mechanized harvester. Journal of Forest Engineering. 1991; 2(2): 25-32.

Owende, P.M.O. 2004. Wood Delivery. In Encyclopedia of Forest Sciences. Volume one. Elsevier Ltd., Oxford. pp. 269-279.

Pickens, J.B., Throop, S.A. & Frendewey, J.O. 1997. Choosing prices to optimally buck hardwood logs with multiple log-length demand restrictions. For. Sci. 43(3): 403-413.

Pnevmaticos, S.M. & Mann, S.H. 1972. Dynamic programming in tree bucking. For. Prod. J. 22(2): 26-30.

Pulkki, R. 1997. Cut-to-length, tree-length or full tree harvesting? Central Woodlands 1(3):22-27,37.

Puolakka, H. 1997. Sumea logiikka käytännön sovelluksissa [Applications in fuzzy logic]. Opetushallitus, Helsinki. 285 p. [In Finnish.]

Puumalainen, J. 1998. Optimal Cross-cutting and Sensitivity Analysis for Various Log Dimension Constraints by Using Dynamic Programming Approach. Scand. J. For. Res. 13: 74-82.

Ramalingham, P. 1976. A branch-and-bound approach to the tree bucking problem. Paper presented at the 1976 National ORSA/TIMS Conference, March 1976, Philadelphia, PA.

Reinders, M.P. 1989. IDEAS, for integral logistics in centralized wood processing. Dissertation no. 1271. Department of mathematics, Wageningen Agricultural University, The Netherlands. 236 p.

Räsänen, T., Imponen, V., Lindroos, J., Malinen, J. & Sorsa, J.-A. 2000. Runkopankki puunhankinnan ohjauksen apuvälineenä [Stem database as a wood procurement planning tool]. Metsätehon raportti 94. 42 p. [In Finnish.]

—, Kuuramaa, I. & Sorsa, J.-A. 2005. Runkopankki ja k-MSN –menetelmä puustotietojen ja pölkkyjakauman ennustamisessa [Applying stem database and k-MSN method to estimating stand characteristics and log output distributions]. Metsätehon raportti 182. 42 p. [In Finnish.]

Schmoldt, D.L., Scheinman, E., Rinnhofer, A. & Occeňa L.G. 2000. Internal log scanning: Research to reality. In: Meyer, D.A. (ed.). Proceedings of the Twenty-eighth Annual Hardwood Symposium: West Virginia Now – The Future For The Hardwood Industry. Memphis, TN. pp 103-114.

Sessions, J. 1988. Making better tree-bucking decisions in the woods. J. For. 86(10): 43-45.

—, Olsen, E. & Garland, J. 1989. Tree bucking for optimal stand value with log allocation constraints. For. Sci. 35(1): 271-276.

Smith, G.W. & Harrell, C. 1961. Linear Programming in Log Production. Forest Prod. J. 11(1): 8-11.

Sondell, J. 1987. Principer och system för datoraptering [Principles and systems for computer-aided bucking]. Forskningsstiftelsen Skorsarbeten. Resultat 6. [In Swedish.]

— & Mitchell, P. 2004. PROMOTE: Production and marketing of timber in Europe. The final report of the EU project FAIR CT 96-1837. Skogforsk, Uppsala. 120 p.

—, Möller, J.J. & Arlinger, J. 2001. Virkesvärdetest 2001 del 2: Aptering - tredje generationens apteringdatorer. Summary: Test of bucking computers. SkogForsk. Resultat 14. 6 p.

Standard for Forest Data and Communication. 2004. Version 2004-10-26. Skogforsk, Uppsala. 98 p.

Strand, L. 1967. Pristabeller for furu ved dynamisk programmering [Price tables for spruce with dynamic programming]. Department of Forest Measurement, Agroc. Univ. of Norway. Melding nr 8: 8-10. [in Norwegian.]

Taylor, B.W. III. 1990. Introduction to management science. 3rd ed. Allyn and Bacon, Boston. 896 p.

Usenius, A. 1986. Optimum bucking of sawlog stems taking the customers' needs into account. Paperi ja Puu. 10: 726-729.

— 1999a. Jalostusketjun kokonaisvaltainen optimointi: Osa 1. Optimointiperiaatteet ja eras sahateollisuuden sovellus [Integrated optimization of the wood raw material conversation chain: Part 1. Optimization principles and an example of an application for the sawmilling industry]. Paperi ja Puu. 2: 114-116. [In Finnish.]

— 1999b. Jalostusketjun kokonaisvaltainen optimointi: Osa 2. Sahan sunnittelujärjestelmä [Integrated optimization of the wood raw material conversion chain: Part 2. Planning system for sawmills]. Paperi ja Puu. 4: 279-284. [In Finnish.]

Uusitalo, J. 1997. Pre-harvest measurement of pine stands for sawing production planning. Acta For. Fenn. 259. 56 p.

— & Kivinen, V.-P. 2000. EMO: A pre-harvest measurement tool for predicting forest composition. In: Usenius, A. & Kari, P. (eds.). Proceedings of the 3rd Workshop: Measuring of wood properties, grades and qualities in the conversion chains and global wood chain optimization. VTT Building Technology, Espoo. pp. 173-180.

— & Kivinen, V.-P. 2001. Factors affecting the output distribution of logs with modern tree bucking optimizing system. In: Sjöström, K. & Rask, L.-O. (eds.). Proceedings of the 2nd World Symposium on Logistics in Forest Sector. Växjö University School of Industrial Engineering and Timber Logistics, Växjö. pp.133-142.

—, Kokko, S., & Kivinen, V.-P. 2004. The Effect of Two Bucking Methods on Scots Pine Lumber Quality. Silva Fennica 38(3): 291-303.

von Essen, I. & Möller, J.J. 1997a. Fördelningsaptering på mindre trakter [Dimension-apportioment merchandising for smaller sites]. SkogForsk. Arbetsrapport Nr 371. 36 p. [In Swedish.]

— & Möller, J.J. 1997b. Fördelningsaptering – en fungerande metod även på smä trakter och vid liten tillåten värdeavvikelse. Summary: Dimension-apportionment merchandising – a useful method even for small sites with minor value deviation. SkogForsk. Resultat 14. 4 p.

Vuorenpää, T., Arminen, P. & Suuriniemi, S. 1999. Arvomatriisien ja tavoitejakaumien laadinta hakkuukoneille [Drawing up value and demand matrices for harvesters]. Metsätehon raportti 81. Metsäteho, Helsinki. 29 p. + 2 appendices. [In Finnish.]

—, Aaltonen, A., Imponen, V. & Lukkarinen, E. 1997. Tukkijakauman ohjaus [Control of log output distributions]. Metsätehon raportti 38. Metsäteho Oy, Helsinki. 24 p. [In Finnish.]

Wang, J., LeDoux, C.B. & McNeel J. 2004. Optimal tree-stem bucking of northeastern species of China. Forest Prod. J. 54(2): 45-52.

—, Xu, S., and Li, G. 1991. Application of network analysis in optimal tree-stem bucking. J. of Northeast Forestry Univ. 19(2): 81-89.

Wolpert, D.H. & Macready, W.G. 1995. No Free Lunch Theorems for Search. Working Papers 95-02-010. The Santa Fe Institute, Santa Fe, NM. 32 p.

Zadeh, L.A. 1965. Fuzzy sets. Information and Control 8(3): 338-353.