Operational Forest Harvest Scheduling Optimisation

A mathematical model and solution strategy

by Stuart Anthony Mitchell

Supervised by Professor David Ryan and Dr Chris Goulding

A thesis submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy at the University of Auckland.

THE UNIVERSITY OF AUCKLAND
NEW ZEALAND

Department of Engineering Science

School of Engineering

University of Auckland

December 15, 2004

Abstract

This thesis describes the Operational Harvest Scheduling (OHS) problem and develops an algorithm that solves instances of the problem. The solution to an OHS problem is an Operational Harvest Schedule (OHS).

An OHS:

- assigns forest harvesting crews to locations within a forest in the short-term (4-8 weeks);
- instructs crews to harvest specific log-types and allocates these log-types to customers;
- maximises profitability while meeting customer demand.

The OHS problem is modelled as a Mixed Integer Linear Program (MILP). The formulation given in this thesis differs significantly from previous literature, especially with regard to the construction of the problem variables. With this novel formulation, the problem can be solved using techniques developed in previous work on aircraft crew scheduling optimisation (Ryan 1992). These techniques include constraint branching and column generation.

The concept of relaxed integer solutions is introduced. A traditional integer solution to the OHS problem will require harvesting crews to move between harvesting locations at the end of a week. However, a relaxed integer solution allows crews to move at any time during a week. This concept allows my OHS model to more effectively model the practical problem.

The OHS model is formulated for New Zealand and Australian commercial forestry operations, though the model could be applied to other intensively managed production forests. Three case studies are developed for two companies. These case studies show improvements in profitability over manual solution methods and a significant improvement in the ability to meet demand restrictions. The optimised solutions increased profit (revenue less harvesting and transportation costs) by between 3-7%, while decreasing the total value of excess or shortfall logs by between 15-86%.

Acknowledgements

I would like to thank Professor David Ryan[1] and Dr Chris Goulding[2] for their supervision of this thesis. Dr Glen Murphy[3], Dr Christine Todoroki[2], Steve Wakelin[2], Dr Hamish Waterer[1], my father James Mitchell and my sister-in-law Sue Turner for taking the time to read through my drafts to correct them.

Thanks to Forest Research limited and the Foundation for Research Science and Technology for providing funding through a Bright Future Enterprise Scholarship.

Thank you to the two companies that provided the information used in the case studies.

The research in this thesis was partially funded by FRST contract C04X0203: "Wood-Fibre Growth and Quality: Tree to Product".

This document is typeset in LaTeX.

[1]University of Auckland.
[2]Forest Research Limited.
[3]Oregon State University.

Contents

List of Tables

List of Figures

List of Algorithms

List of Acronyms

AC . . . . Australian Company One of the two companies that provided case studies.

B&B . . . . Branch and Bound A technique for finding integer solutions from a RLP.

CA . . . . Crew Allocation The decision to place a particular crew in a harvest unit in a period, also the constraints that model this decision in the OHS formulation.

CGA . . . . Column Generation Algorithm The sub-problem in Column generation.

DP . . . . Dynamic Program A technique used to solve some types of OR problems.

FOLPI . . . . Forestry Orientated Linear Programming Interpreter Estate modelling tool created at *forest research* and commonly used in NZ.

forest research . New Zealand Forest Research Limited The New Zealand Crown Research Institute responsible for research in forestry.

FSC . . . . Forest Stewardship Council An organisation that encourages sustainable forestry.

GIS . . . . Geographical Information System

GSPP . . . . Generalised Set Partitioning Problem An extension of the SPP.

GUB . . . . Generalised Upper Bound A term used to refer to a type of constraint common in scheduling problems.

IP . . . . Integer Program An extension of an LP which only contains integer variables.

LP . . . . Linear Program A technique used to solve some types of OR problems.

MARVL . . . . Method of Assessment of Recoverable Volume by Log-type Standing inventory tool created at *forest research* and commonly used in NZ.

MILP . . . . Mixed Integer Linear Program An extension of a LP to include integer variables.

NFP . . . . Network Flow Problem A special subset of LP problems that will give naturally integer solutions.

NZ . . . . New Zealand

NZC . . . . New Zealand Company One of the two companies that provided case studies.

OHS . . . . Operational Harvest Scheduling Short term scheduling of forestry crews within a forest with production allocation.

OHSA . . . . Operational Harvest Scheduling Algorithm The algorithm that solves the OHS problem.

OR . . . . Operations Research The science of better.

PLE . . . . Probable Limits of Error The confidence interval of an estimate expressed as a percentage of the mean.

PT . . . . Production/ Transportation The constraints that model the linear transportation, and production decisions in the OHS formulation.

RLP . . . . Relaxed Linear Program The problem resulting when the integer restrictions on the MILP are removed, and the MILP solved as a standard LP problem.

RMP . . . . Restricted Master Problem The master problem in Column generation.

SED . . . . Small End Diameter The diameter of the smallest end of a log.

SPP . . . . Set Partitioning Problem A special type of IP.

ZIP . . . . Zero-one Integer Programming A programming framework developed at the University of Auckland (Ryan 1980), for the solution of large scheduling problems.

Chapter 1

Introduction

In Xanadu did Kubla Khan
A stately pleasure dome decree:
(Coleridge 1798)

Forestry science is the study of how to harvest trees, use forest resources, and plan the harvest of the forests to satisfy long-term goals. Modern forestry seeks to provide sustainable constant high-quality production, while preserving ecological values. The importance of forestry can only grow with the increase in worldwide interest in sustainable resources and climate change.

This thesis is concerned with the planning of short-term forest harvesting operations. This planning is called Operational Harvest Scheduling (OHS). The Operational Harvest Schedule OHS assigns forest harvesting crews to locations in a forest in the short-term (4-8 weeks). These crews are also given instructions detailing which logs to harvest and which customers to supply. The allocation of logs should satisfy customer orders while maximising profitability.

The next section looks at the motivations to produce better OHSs .

1.1 Motivation

Harvesting a large forest is a complex operation. There may be a number of harvesting crews to schedule, and each tree in the forest can be used to make several different types of log. The types and volumes of logs produced from any location within a forest are governed by the size and quality of the trees found in that location. The harvested logs are used to satisfy various orders from customers, located in a variety of distant places. A good operational harvest schedule will satisfy the customers' orders from areas of forest while minimising transportation costs. Also, areas of the forest with high quality trees will be used to provide high quality (high-value) logs while lower quality areas will fulfill low quality (low-value) demands, minimising the conversion of high quality trees into low-value logs.

Operational harvest scheduling can be very difficult. A poor harvest schedule may have some or all of the characteristics listed below.

- The crews move more often than necessary.
- Large stockpiles of unsold logs are created.
- High quality trees are harvested to make low-value logs.
- There are large transportation costs; because logs are delivered to customers that are located far from the area where the trees are harvested.
- Customers are unhappy; because they do not receive the volume of logs that they have ordered.

Good operational harvest scheduling is an activity where large immediate financial gains can be found. This is in contrast to tactical planning where gains will be seen in two to three years, or strategic planning with gains realised thirty or more years in the future (Chapter 2 describes the hierarchy of plans). The immediate gains of a good harvest schedule are listed below.

- Harvesting costs are reduced.
- Harvesting productivity is increased.
- Larger volumes of high quality logs are produced.
- Transport costs are lowered.
- There is a decreased need to purchase volume from other sources to supply customers.
- The crew movement costs are reduced.

These potential gains have generated considerable interest in OHS optimisation from the forestry industry. Other gains from good operational harvest scheduling derive from the ability to accurately predict future harvesting volumes. This implies that sales people can accurately predict if there is enough volume to meet current orders, or analyse the effects of selling large volumes of low-value logs versus smaller volumes of high-value logs. Harvest schedulers can also anticipate, in advance, the resources (harvesting crews, logging trucks, etc) needed to harvest the forest and meet customer demands.

1.2 Chapter description

The details discussed within each chapter are described below.

Chapter 2 discusses the forestry background necessary to understand the OHS problem. This includes a history of New Zealand forestry practice; silvicultural practices; methods of resource assessment; forest harvesting techniques; the nature and usage of the logs produced; and the place of the OHS plan within the forestry planning hierarchy. This background will give the reader an understanding of both forestry terminology, and techniques used in the thesis.

Chapter 3 expands upon the background given in Chapter 2 and describes the processes and terminology that must be understood before a OHS model can be formulated. This chapter describes the decisions that are found within an OHS plan; the nature of the harvest units considered; the challenges and opportunities presented by generating new yield predictions; various considerations that stem from the physical harvesting of trees; and the nature of the market which buys logs.

Chapter 4 reviews the literature in the forestry-planning field. The different time horizons in forestry planning are discussed along with the interrelationship between the different plans within the planning hierarchy. This chapter examines a number of different optimisation techniques used to solve forestry planning problems in the literature. Other OHS solution algorithms given in the literature are examined in detail, including both formulations, and solution methods used.

Chapter 5 discusses the Linear Program (LP), Mixed Integer Linear Program (MILP), and Set Partitioning Problem (SPP) frameworks that are used in this thesis to construct the mathematical programming formulation of the OHS planning problem. Following the discussion of these frameworks, this chapter briefly describes the algorithms used to find the solutions. These algorithms include the Simplex algorithm, Column generation, and Branch and Bound. Though full discussions of these algorithms can be found elsewhere (for example Bazzaraa et al. (1990)), this chapter will prepare the reader for the concepts used later in this thesis in the discussions of the problem formulation and solution strategy.

Chapter 6 shows the specific formulation of the model used in this thesis to solve the OHS problem. This chapter evaluates a number of different ways that the model may be formulated, discussing in depth the formulation traditionally used in literature, and contrasting it with the formulation found in this thesis. This chapter describes the various constraints in the formulation and divides them into two sections, namely the Crew Allocation (CA) and Production/ Transportation (PT) sub-models. The similarity between the Crew Allocation (CA) sub-model and a Generalised Set Partitioning Problem (GSPP) is discussed. The constraints are detailed and the construction of the variables and the objective function is given.

Chapter 7 describes the extensions to the traditional MILP formulation used in this thesis. The discussion focuses on the relaxation of the integer requirements for a solution. The solution to the OHS plan does not require any variables to be strictly integer, but rather requires that groups of variables that represent a single crew must be compatible and provide an unambiguous interpretation of the crew's location through time. This chapter also details the extra requirements placed on the construction of variables that represent a sequence of work for a crew. These restrictions ensure that the difference between the Relaxed Linear Program (RLP) solution and the eventual solution is reduced.

Chapter 8 details the specific implementation of the techniques shown in Chapter 5 used to solve the formulation of the model. Chapter 8 describes: the use of ZIP (Ryan 1980); the solution of the RLP problem; the details of the specific column generation algorithm used to generate the crew schedules variables; the branch and bound process that finds integer solutions that fit the specific definition of the integer solutions; the integer allocation heuristics that are used to quickly find solutions to the OHS; the generation of new yield predictions for harvest units as the solution algorithm progresses; and the cause and resolution of end-effects in the OHS.

Chapter 9 describes the three case studies that were considered in this thesis. The case studies involve an Australian and a New Zealand company. These case studies were used to successfully validate the formulation and solution strategies in the thesis; the results were found in short time and reasonable OHS solutions were obtained. In two of the case studies, a comparison of the manual solution to the OHS solution is provided.

Chapter 10, discusses the outcome of the research contained in this thesis. It looks at the issues of problem formulation; yield prediction generation; the concept of relaxed integer solutions; the solution strategies needed to find these types of solution and the results of the case studies.

Chapter 2

Forestry Background

Where Alph the sacred river, ran
Through caverns measureless to man
Down to a sunless sea.
(Coleridge 1798)

This chapter provides some background to the Operational Harvest Scheduling (OHS) problem. A brief background of plantation forestry in New Zealand is presented detailing its history and current practice. The techniques used in forestry planning are then discussed.

2.1 History

Purey-Cust & Hammond (1995) states that the history of New Zealand forestry is divided into three major planting events:

- the first circa 1920-1935;
- the second from 1960-1984
- the third from 1991 to the present day[1].

This paper identifies that the causes of the first *planting boom* were public concern for the wasteful forestry practices of the day, and recognition of the inability of native timbers to supply a sustainable resource for New Zealand's future needs.

[1]Planting has dropped from its 1995 peak but it is still relatively high (M.A.F. 2002).

During the first boom, *Pinus radiata* was gradually accepted as the best exotic species to grow for New Zealand's general timber needs. Most forests planted at this time were located on the pumice plateau in the central North Island. The trees planted in this era are commonly known as the *Old crop*. There are still a few stands of this era that have been preserved for historical purposes.

The second planting boom in the 1960s was a response to the identification of forestry as an important export industry for New Zealand, and a realisation that more planting was needed to support it. This planting, by both the government and private interests, was distributed throughout New Zealand and not concentrated in the central North Island as before. In contrast to the Old crop, improved silvicultural practices were used, such as pruning and thinning, along with the gradual phase-in of tree stocks improved by selective breeding.

From 1991 onwards, forestry has been a growth industry in New Zealand with new plantings throughout the country. Small private investors wishing to provide for their superannuation and recognising the potential of forestry as a future industry drive this new planting.

The conversion of new land to forestry is attributed to the very competitive comparative returns of forestry versus traditional pastoral land uses, such as drystock or dairy farming. Forestry has become widely recognised as a suitable use for otherwise marginal farming land.

2.1.1 Current situation

Production forests in New Zealand are mostly plantations of exotic species. The primary focus of silvicultural and forestry practices is to increase profitability. This is in contrast to New Zealand's treatment of its indigenous forests, which are managed for conservation and recreational uses. Wood production from indigenous forests in New Zealand is currently very small. It was limited to 100,000 m^3 a year as at 2000 (M.A.F. 2002), compared with 18 million m^3 from plantation forests. The volume harvested from indigenous forests (especially on government owned land) continues to decrease due to sustainability requirements in New Zealand law and current strongly opposed public opinion.

Current New Zealand forestry practices are described below.

2.2 Silviculture

There are a large number of possible combinations of silvicultural practices used to obtain a final crop. The silvicultural treatments used on a tree throughout its life are known as a *regime*. Differences between regimes include the type of seedlings planted, the initial stocking, thinning choices and pruning practices. Each regime will produce different volumes and products when the trees are clearfelled. The choice between regimes is determined by the required outcome, the initial investment of the forest owner, the site where the trees are located, and personal preferences.

The management practices of plantation forestry in New Zealand are based on the intensive, even-age, monoculture of *Pinus radiata*. When harvesting, a section of forest containing 20-30 year old trees (determined by the regime) is clearfelled and then replanted with new seedlings that are genetically improved through selective breeding. The silvicultural regime of the forest is commonly targeted to grow high-value timber products for export, so the trees are:

- planted in medium to low stockings;
- pruned to produce high-value butt logs;
- sometimes production thinned.

These practices are detailed in Maclaren (1993), which is a comprehensive manual on current New Zealand forestry practices.

2.2.1 Planting

The number of seedlings planted per hectare is dependent on the type of regime chosen, and is known as the *initial stocking*. Common initial stockings are between 400-2000 trees/ha (Maclaren 1993). The ground is cleared before planting by burning or by the use of herbicides. Seedlings are commonly planted in an approximately square grid of rows and columns.

2.2.2 Thinning

Thinning is the removal of trees before the final harvesting. Often the initial stocking is greater than the *final stocking* (the stocking at harvest time). The difference between the two is achieved through natural losses and thinning. Trees that are chosen for thinning are usually inferior: either small or malformed.

Thinning can occur at any age. If the trees are removed early (first third of the rotation) and no logs are harvested, it is called thinning to waste. *Production thinning* occurs later in the rotation when the trees are of a marketable size, and the stems may be sold to customers. Production thinning is discussed in more detail in Section 2.5.1.3.

2.2.3 Pruning

Pruning is the removal of the lower branches from young trees to increase the quality and value of the logs produced. The removal of young branches produces *clearwood* (knot free wood), in the *pruned butt* log (the bottom-most log produced). Clearwood is suitable for appearance grade uses as it is free of knots.

Pruning usually occurs in several stages (lifts), from about three to ten years after planting. These stages are timed so that the branches are trimmed early enough to maximise clearwood while still allowing the tree to grow well.

In NZ, nearly 70% of the final harvest has been pruned (M.A.F. 2002).

2.3 Area description

In order for forest planning to be successful, there must be a way of describing the existing forest. The first step in this process is to segregate specific areas of forest that share common properties.

Three ways to classify areas of forest are by:

- crop-types;
- stands;
- harvest units.

The classification used depends on the needs of the application.

Crop-types are groups of areas that share the same management regime. These smaller areas are distributed throughout the forest. Therefore, a crop-type is not a single contiguous piece of forest. Crop-types are used in strategic planning (see Section 2.7.1). *Age-classes* result when a crop-type is divided based on age.

Stands, or alternatively compartments, are areas of forest that are homogenous with respect to future management. Unlike crop-types, stands do have a single physical location. Often stands are used in tactical planning (see Section 2.7.1) and when inventory measurements are planned (see Section 2.4).

Harvest units are defined in this thesis as a specific area of forest that can be harvested in a single operation. The boundaries of each harvest unit are defined in the harvest planning process. Harvest planning determines the harvesting operations (see Section 3.4) needed in that area. A harvest unit is also known as a setting or block (especially in steeper areas). In flat areas, a harvest unit will be almost identical to a stand or compartment. A harvesting crew is able to complete a harvest unit without movement or set-up penalties (discussed in Section 3.4).

Harvest units are derived from the stands used in tactical planning. Thus, a harvest unit should be homogeneous throughout its area and the yields in any part of the harvest unit should approximate the harvest unit average. In addition, each part of the harvest unit is able to be harvested using the equipment and harvesting crews specified for the whole harvest unit.

Once these areas are identified, the next step in the planning process is to find an estimate of the future production of these areas.

2.4 Yield prediction

2.4.1 Some definitions

This section discusses various processes that are used to estimate the future production of a forest. To describe these processes it is useful to define some concepts and terms.

- *Tree*: A tree within a forest before it is felled.
- *Stem*: The section of the tree above ground without branches, also refers to a fallen tree with its branches removed.
- *Stem-description*: A physical description of the dimensions and quality of a stem.
- *Logs*: Sections of stem. Usually cut to sell to customers.

- *Log-type*: A specific type of log with quality and dimensional specifications, for a particular customer and/or use.
- *Timber*: Boards, fittings etc., that are the products of sawmills.
- *Yield prediction*: A specific estimate of the volume by log-type that will result when a harvest unit is clearfelled.
- *Standing Inventory*: Or simply *inventory*, is the trees within a forest before they are felled.
- *Cruising*: The collection of physical data from the trees within an area, typically by skilled forestry workers who walk through the forest and measure trees.

2.4.2 Inventory

Some method of predicting future yields from the forest is needed for successful planning. The accuracy required from these predictions depends on their usage. For instance, estimated yield predictions for aggregated log groups are used in long-term planning. These long-term aggregate estimates could be based solely on the management regime. In the shorter-term other, more accurate, methods need to be used. Methods used to find pre-harvest inventory used in the OHS problem and tactical planning are described below.

A forestry company will use a specialist inventory software package to collate, calculate and present standing inventory estimates. The base data for these estimates are found by cruising an area of forest. Statistical functions in the inventory package, along with the sampling techniques used while cruising, allow a sample of trees to be cruised to obtain an estimate of the yield of the whole harvest unit. This practice reduces the expense of collecting inventory measurements.

Inventory packages can also include models that simulate tree growth. These growth functions can be used to extend the usefulness of inventory measurement, as accurate current yield predictions can be based on inventory measurements made in the past. The inventory package will also contain some method to calculate the volume of log-types from the cruised information.

2.4.3 Sampling techniques

In most commercial forests, there are too many trees to allow the cruising of every tree considered in the inventory. Therefore, sampling techniques are applied so that the entire inventory can be approximated from the data recorded from a smaller number of cruised trees. The sampling method recommended is to pick sample plots separated spatially and measure only trees within the plots.

Inventory plot selection can be random, systematic or stratified. Random plot selection consists of locations taken totally at random; systematic schemes place the plots on a regular grid; while stratified schemes use available information to break the region into homogeneous areas. These regions could be homogeneous because of the silvicultural technique or site-specific factors such as slope. The actual trees sampled in the plot can then be chosen by a number of different methods. These methods determine whether all the trees in the plot or a subset are sampled.

2.4.4 Growth and other models

Growth models are used to predict the current standing inventory when inventory measurements were taken some time in the past. These models account for the growth of the trees in the time between inventory measurement and the eventual harvesting.

Other models that can be used to generate more accurate yield predictions listed below.

- Taper equations estimate the shape of the stem.
- Bark equations estimate the thickness of the bark.
- Breakage equations estimate if and where a stem will break when it is felled.

2.4.5 Yield estimation

Yield estimates for a harvest unit can be generated from three different types of data collected within the forest.

- Past yield information for similar harvest units is applied to the current harvest units.
- An estimate of log-type volumes is found directly from observation, before harvest, of sampled stems in the harvest unit.
- Inventory software simulates the bucking process to determine yield predictions from stem descriptions.

Past information and log-type volume predictions provide log volume data directly and give fixed estimates of yield. These estimates are based on decisions made at the time of the inventory measurements. It is possible to include some flexibility in these estimates by allowing conversion between log-types.

A stem description must be converted into log volumes. A *Bucking algorithm* allows this by simulating the log making process (see Section 2.5.1.4) on each stem. The bucking algorithm is applied to each sampled stem in the inventory. It allocates feasible log-types along the length of the stem. The total estimated yield of the harvest unit is then found by aggregating the log volumes produced from each of the sampled stems.

A bucking algorithm provides flexibility as several possible yields can be generated for a single harvest unit. Each yield prediction will represent a different utilization of the resource. The literature on Bucking Optimisation (Section 4.2.1) discusses in depth the use of this ability to change yields.

There are three main types of yield prediction algorithm.

- Fixed yield estimates may be altered by log conversions.
- A bucker can optimise value with a Dynamic Program (DP) recursion.
- A bucker can use a priority list heuristic.

2.4.5.1 Log conversions

Past yields and direct estimation of log volumes give fixed estimates of yield. Log conversion rules and factors can introduce flexibility into these yield predictions. It is assumed that some logs or groups of logs are interchangeable, thus some or all of the volume allocated to one log can be converted into volume of another. Typically, the conversion rules allow volume from high quality logs to be downgraded into lower quality products. This flexibility allows new yield predictions to be created from the fixed estimates.

2.4.5.2 Dynamic programming bucker

A Dynamic Program (DP) bucker (see Section 5.2.2.1) finds the optimal value of each stem given a stem description and a description and price for each log product. Common implementations of these buckers divide the stem into stages along its length. A DP recursion then solves a longest path problem, to maximise the value of the stem. This is equivalent to a shortest path problem with the objective multiplied by -1 (note there is no possibility of a negative length cycle). The solution to this shortest path problem will allocate logs along the stem. Figure 2.1 shows one possible way to buck a stem into three different logs. See Deadman & Goulding (1979) for a complete reference.

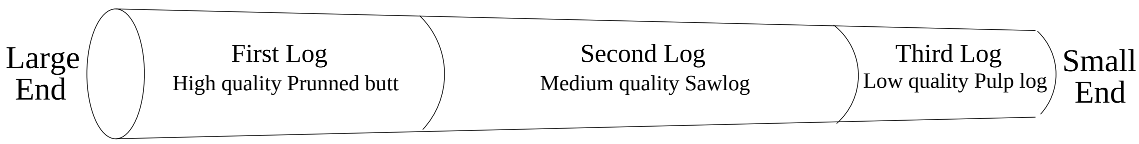


Figure 2.1: A bucked stem

2.4.5.3 Priority list bucker

A priority list is a common implementation of a non-optimal bucking heuristic. For a detailed description, see Laroze & Greber (1997). Instead of optimising value over the entire stem, logs are allocated to each section of the stem using a priority list (an ordered list of possible logs), without looking ahead. From the base of the stem the bucker attempts to fit logs from the highest priority log to the lowest. The bucker then repeats the process from the end of the first log, to find the second log until the end of the stem is reached. There can be various slight alterations of this process to give a better solution especially in the allocation of the last two logs. These alterations can involve looking ahead so that the wastage at the end of the log is minimised.

2.5 Harvesting

After the planning process is completed, the forest is harvested (hopefully according to the plan). The extraction of logs from a harvest unit is accomplished by *harvesting crews* (or gangs). The operations, composition, and size of a harvesting crew can vary greatly. Crews can *clearfell* a harvest unit, where they harvest all the stems in the unit and leave bare land for replanting. Alternatively, the crews can do *production thinning*, where only certain trees are selected for extraction, while the rest are left for future harvesting. Small crews, in thinning operations, can have a production rate of $50\mathrm{m}^3$ of logs per day, while larger highly mechanized crews, in clearfelling operations, can produce up to $1000\mathrm{m}^3$ per day.

2.5.1 Crew operations

In New Zealand crews can be characterised by their type of extraction operation:

- *Ground-based crews* use *forwarders*, *skidders* or *bulldozers* to move the stems once they have been felled. These machines move the stems along the ground either by dragging them with chains or by picking up the stems.
- *Cable crews* use cables and towers to extract stems by attaching them to long winch ropes.

Ground-based operations tend to be cheaper and more productive. However, they are not able to handle steeper[2] harvest units. On the steeper slopes more expensive and less efficient cable logging operations are used. These operations also have less impact on the soil in the harvest unit than ground-based operations. The harvest unit topology and soil characteristics determine which harvest units a particular crew may harvest and its production capacity and cost.

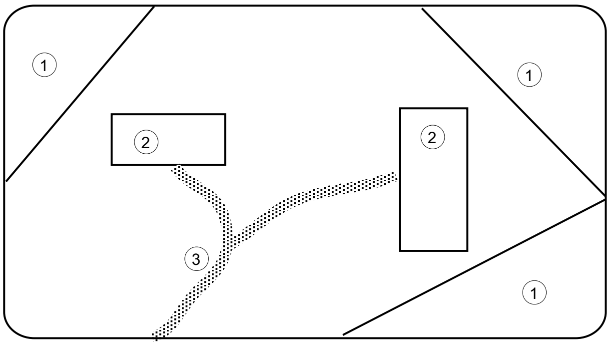
The operation of the crews can be separated into felling, stem extraction and landing operations. The landing operations are similar in both ground-based and cable logging operations. Felling and stem extraction operations distinguish the two.

2.5.1.1 Ground-based operations

In ground-based operations, the crews can be highly mechanized and production volumes can be very high. The large degree of mechanization is possible because the harvest units are on relatively flat country, so large wheeled or tracked machines can be used, as opposed to steeper slopes where the ground and soil will not allow these machines to operate.

Figure 2.2 is a schematic diagram of the operational layout for a ground-based operation. Trees are felled and delimbed (branches are removed) at the felling faces, possibly by large *mechanical fellers* (machines that cut down trees). Then they are dragged to the landings by the skidders. These large rubber wheeled vehicles move the stems with chains and winch ropes or by hydraulic grippers.

[2] Slopes consistently over 25° are considered steep in New Zealand.



1. Felling faces
2. Landings
3. Road

Figure 2.2: The basic layout of a ground-based harvest unit

2.5.1.2 Cable logging

Cable-logging operations are used when the ground slope is too steep for ground-based operations. On steeper ground skidder operations cannot be used to extract the stems from the felling face. Either it will be too dangerous to use a skidder, as it will be in danger of rolling, or the damage to the soil in the harvest area will be too great if enough earth is moved to make skidder operations safe.

Fallers (forestry workers that cut down trees manually) on the ground usually carry out felling in a cable logging operation. Mechanized fellers are not used because they would cause too much damage to steep slopes. A tower and cable set-up is then used to extract the logs above the ground to the landing. A bird's-eye view of the operation is shown in Figure 2.3.

2.5.1.3 Production thinning

In *production thinning* operations, a small crew enters a harvest unit perhaps several years before it is due to be clearfelled, and removes selected trees. The stems are usually chosen to improve the overall value and condition of the residual trees in the harvest unit. Due to the quality and size of the stems removed most of the volume in production thinning goes into pulp logs, though some sawlogs can be recovered.

Almost all production thinning takes place in harvest units that are suitable for ground-

1. Opposite Ridge
2. Hauler
3. Fallen stems
4. New felling face
5. Landing

Figure 2.3: Cable logging layout

based harvesting, as the entry costs to a cable harvest unit are too expensive to make production thinning economical. Production thinning is similar to the ground based operations described in Section 2.5.1.1, using smaller versions of the machines used for clearfelling.

2.5.1.4 Landing operations

The *landing* is an area of flat cleared land where bucking occurs and logs are stacked before they are loaded onto trucks.

The size and number of landings in a harvest unit determine the number of crews able to operate simultaneously in the harvest unit. In most harvest units, only a single crew is able to operate at any one time. However, in larger harvest units suitable for ground based harvesting, two or more crews can operate simultaneously. The harvest planning process (Section 3.2.1) will determine how many crews can work in each harvest unit.

After stems are moved to the landing, a *log-maker* decides how each stem should be bucked to create the logs required. The log-maker measures the physical dimensions of the stem and assesses its quality characteristics (see Section 2.6.2). These physical and quality characteristics are checked against the allowable characteristics of the various logs to be made from the stem. The value recovered from each stem is heavily dependent on the talents of the log maker. A good log-maker is able to make decisions that maximise the value obtained from each stem. A poor log-maker will fall short of this ideal.

Parker et al. (1995) has found the efficiency of log-makers also decreases as the number of log-types considered increases. As the number of log-types considered increases above ten significantly more errors are made classifying the stem sections.

Because of the variability of log-makers' skills, various products exist that can assist the log-maker. A system of particular interest is the *IFRLogger* ™(*IFR Logger* 2002) system (shown in Figure 2.4). This system consists of electronic callipers that can measure the diameter of the stem at several positions along its length, together with a system to enter log quality information into the callipers. A dynamic programming algorithm then determines the sequence of cuts required to obtain optimal value from the stem. The log-maker marks on the stem the positions of the cuts that are to be made by skid workers and labels the resultant logs with their grade.

Figure 2.4: The use of IFRLogger in log making

The skid workers use chainsaws to buck the stems into the logs at the points indicated by the log-maker.

The jobs of the feller, log-maker and the skid worker can be combined in a single mechanized harvesting machine (for example, the Waratah harvesting heads[3]). These machines scan the characteristics of the log and use the results of a computer heuristic to determine how to buck it. Unfortunately, due to high stem variability in New Zealand and Australia, mechanized harvesting machines can deliver unsatisfactory value recovery if there are many log-type with tight specifications. However, this may change with improvements in stem scanning technology,

[3]http://www.waratah.net

and better predictive algorithms to inform the bucker.

After the logs are made, loaders pick up the logs and stack each type of log into a different pile. The number and size of these piles limit the number of log-types that can be harvested at one time. If too many different log-types are made at a single landing, it will become too crowded as the log piles will take up all the available space. As trucks come during the day, logs are placed on the trucks by the loaders and taken to the customer. Figure 2.5 shows a loader stacking logs.

Figure 2.5: A loader operating

2.5.1.5 Super-skids and central processing yards

Alternatives to the landings described above include *super-skids* and *central processing yards*. Both of these options remove the landing operations from each individual harvest unit and transport whole stems from harvest unit to a central location, where they are bucked. A super-skid can be significantly larger then a normal landing, perhaps up to ten kilometres away from the harvest units, and able to service several harvest units.

Central processing yards are used by FCF (Fletcher Challenge Forests) in Kaingaroa forest in the central North Island of New Zealand. In this forest, stems from a large part of the forest are transported to one of two central processing yards, where the logs are made, then transported to customers. One of these yards, the highly mechanized KPP (Kaiangaroa processing plant) processes 2,000 stems a day, one third of the total production of Kaiangaroa forest. The other yard is located near the railhead in Murupara. At this yard, stems are bucked manually.

The double handling cost of this method is justified by several advantages. A greater efficiency is achieved because the landing operations are concentrated in one place. An increase in value recovery is possible when all the stems are bucked centrally. Large numbers of log-types can be produced simultaneously from stems obtained throughout the forest.

In order for this approach to be viable, vehicles that are capable of moving whole stems are needed. These vehicles are very long as the stems can be in excess of thirty metres. In New Zealand, these trucks are able to operate only on private forestry roads.

The logs bucked from the stems are destined for different customers. These customers have various uses for the logs and require them to meet different specifications. The usages and specifications are discussed below.

2.6 Log usage and specifications

The aim of a commercial forestry operation is to sell the logs it extracts from the forest to maximise revenue and profit. There are many different types of logs made, due to the variety of end uses of wood. The number of different uses is matched by the variation in wood quality and dimension that is a result of the natural process of tree growth.

2.6.1 Log usage

Logs can be grouped by their end usage. These groupings are:

- Sawlogs;
- Peeler logs;
- Pulp (or Chips);
- Poles.

Sawlogs are used to supply local sawmills or are exported to sawmills overseas. Sawlogs usually need to meet strict quality and size requirements. These requirements exist because of the type of timber products that are made from the logs. There are several different grades of sawlog and the larger, straighter and most defect free logs attract the highest values.

Peeler logs are required to be straight and long. Other quality requirements are similar to saw logs. When processed, peelers are rotary peeled on a lathe. A thin layer of wood is cut from the surface, as a paper towel is pulled from a roll. These logs are used to make veneer and plywood. The end use of peeler logs varies from high quality defect free veneer to industrial plywood, which may have defects.

Pulpwood satisfies the paper industry requirements for wood fibre used to manufacture paper products. As the wood is made into pulp, the logs are of a lower quality than sawlogs. The physical characteristics of pulp logs are based on ease of handling and the ability of the pulp mill to process the logs provided. Pulp mills are volume orientated and pulp is the largest single log-type, at around 24% (M.A.F. 2002) of the total volume extracted from New Zealand forests. Pulp logs are both the lowest quality log grade and the lowest priced logs. Forestry companies are often locked into long-term supply contracts to the pulp customers. Unfortunately, if the forestry company is currently harvesting areas with good quality timber the need to meet these long-term commitments can require high-value resource to be degraded and sold as low quality pulp logs.

Poles are a specialised end use and are used to make products including telephone poles and fence posts. Pole logs are usually extracted from thinning operations and are required to be tall, narrow and straight.

2.6.2 Log specifications

The specification of logs depends on the particular methods used in the company. Any one log-type can be classified by a combination of any of these properties:

- quality or grade;
- length and diameters;
- end usage;
- intended customers.

2.6.2.1 Log quality

Log quality affects the suitability of timber for its end uses. Log quality can be roughly classified into these groups:

- pruned butt;
- internodal;
- long straight logs, with small branches;
- short straight logs, with small branches;
- swept logs, large branches, out of round etc..

The pruned butt is found at the base of the stem and is the largest log. Pruned trees (see Section 2.2.3) have had the branches removed up to six meters in height. This results in no branches on the logs and very few defects in the timber. The pruned butt is suitable for high-value logs. The demand for clear wood (without defects) sawlogs is driven by requirements for appearance grade products. Appearance grade products are used for mouldings or other uses. Knots and defects caused by branches will give an unattractive final product.

Internodal logs contain long sections of clear wood between clusters of branches. These sections of clear wood are suitable for high-value products.

Straight logs with small branches are suitable for medium value products and can increase their value with the use of finger-jointing or other technologies in the sawmill. These logs can be used for timber framing or other structural uses.

Sweep, large branches, severe ovality or other defects cause a log to be unsuitable for all but low-value products, boxwood for example. These logs are often sold as pulp, or depending on the degree of defects they may be left as waste.

2.6.2.2 Length and diameters

The length of a log is simply the distance from one end to the other. The two important diameter measurements are the small and large end diameters (SED, LED) of the log. Figure 2.6 shows the measurements of an example log and some quality features that are present.

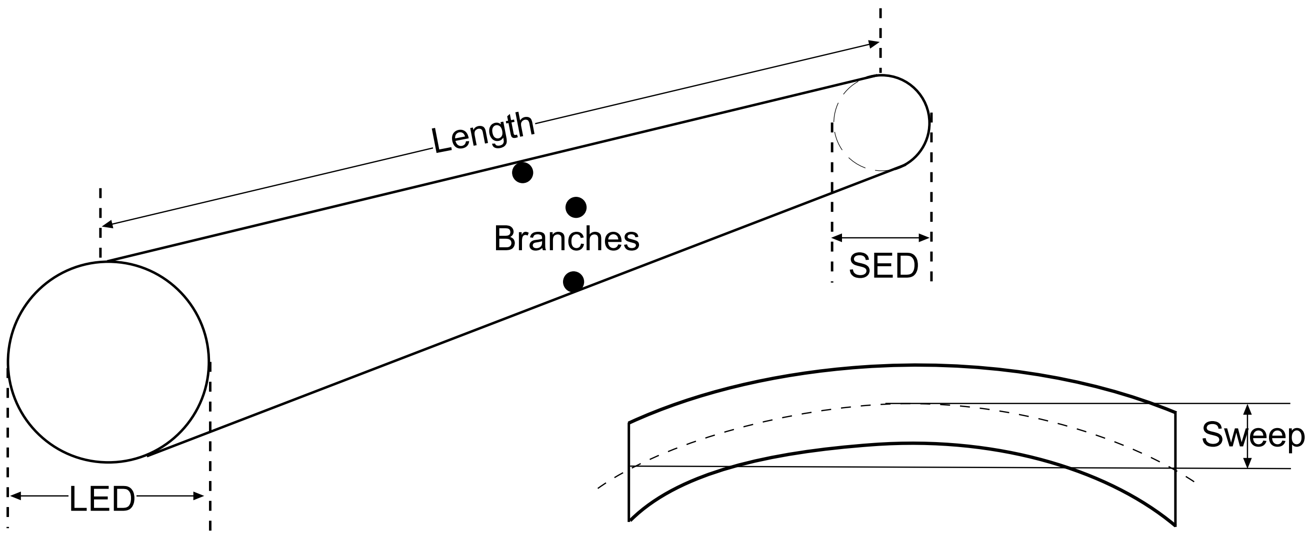


Figure 2.6: Various measurements used in a log description

2.6.2.3 Intended customer

The intended customer of a log is often an important factor as the specifications for logs can vary between customers. This has the unfortunate effect that often very similar logs are not interchangeable because of a specific requirements differing between customers.

2.6.3 Aggregate requirements

Often, as well as specifying the permissible range of values for the log measurements, the customer may specify that the total order has some aggregate average measurement. This specification method is often used when there is a wide range of allowable dimensions in the log specification. The average requirement ensures that the customer does not receive a shipment of logs that meet minimum requirements with no logs in the middle or higher part of the acceptable range. In particular, it is common to specify an average SED requirement for an order of logs.

Some customers may require a total volume of a specific grade (with minimum and maximum lengths) but allow the lengths supplied to vary. The proportion of different lengths within the order can be given as a range. For instance, an order of export logs may contain both export long logs and export short logs. The total volume for both these logs together may be fixed while the export longs may be constrained to make up at least 25% of the final volume, and not more then 75%. Table 2.1 shows some of the common log grades in New Zealand.

Table 2.1: Common New Zealand log grades[4]

Destination	Log-type	Grade	Min SED (cm)	Max Branch (cm)	Length (m)	Percentage Allowed
Domestic	Pruned Sawlogs	P1	40	None	3 to 6	
		P2	30	None	3 to 6	
	Domestic Sawlog 1	S1,S2	25	7	3 to 6	
	Domestic Sawlog 2	L1,L2,L3	16	14	3 to 6	
	Chip logs (pulp)	R	8		3 to 8	
Export	Japanese ‘A’	AL	30	10	12.1	Min 70%
		AM	30	10	8.1	Max 30%
	Japanese ‘J’	J	20	16	As for ‘A’	
	Korean ‘K’	KL	20	20	11.1	Min 60%
		KM	20	20	7.4	
		KS	20	20	5.5	Max 15%
	China ‘C’	C	18	20	10,8,6,4	

Notes:

- The percentages relate to the proportion of the grades in an export consignment of logs. Meeting the high percentage of long logs is the main difficulty, and therefore longer logs have a premium value.
- Lengths for Japanese A (or N) can include 6 metre or 4 metre with appropriate overcut of trim allowance.
- Korean KM can be down to 15cm SED for specific shipments, while lengths for K can also cover 7.4m or 3.7m.
- US grades usually approximate K grades.

2.6.4 Log price

In the forest industry there are, unfortunately, a large number of different pricing arrangements. These arrangements depend on the business relationship between the customer and the supplier. The number and variety of these arrangements makes comparing the relative value of logs very difficult, and can lead to contracts to supply logs that actually lose money for the forestry company. In general the pricing arrangements are $/$\mathrm{m}^3$ or $/tonne. They differ in which costs of harvesting and transportation are borne by the supplier, and which costs are borne by the customer. Different possible arrangements are listed below.

- *Stumpage (or Royalty) sales*- The rights to harvest the trees are sold. Harvesting and transport costs are paid for separately by the customer. Sometimes customers perform the harvesting and transport themselves. This situation can lead to complex *Buy back* arrangements where the customer only requires pulpwood and not saw logs or vice versa.

[4]Table and notes taken in full from Hipkins (1995).

- *Roadside sales*- The harvesting costs are paid by the supplier and the customer pays the transportation costs.

- *Forest gate sales*- The transport costs are split between the supplier and customer. The supplier pays the portion to the edge of the forest while the customer covers the rest of the transportation cost separately.

- *Delivered sales*- The customer receives the logs and pays a single price while the supplier pays all of the harvesting and transportation costs.

2.7 Forestry planning

We have now briefly discussed New Zealand forestry practices from planting a tree to delivering logs to customers. Forestry planning overviews and directs these practices. Forestry is a large primary production industry that requires large amounts of capital investment and a long-term outlook. Large capital investments are needed for timber harvesting machinery, timber processing plants and the transportation of products. The investments in tree planting, silviculture, and the opportunity cost of the land used are not recouped until the trees reach maturity[5]. At harvest, a wide variety of logs are produced because of natural variability in tree growth. These logs undergo a range of processing, by a large number of different customers (who can be located far from the forest itself). The customers will have constraints on the volume they can accept and will be willing to pay a range of prices. The combination of all the above factors has given the discipline of Operations Research (OR) much to offer the forestry industry.

Modern forestry has now come to depend on quantitative analysis in all long-term forest planning exercises. An invited review of challenges for OR practitioners, Martell et al. (1998) comments that “OR has unquestionably influenced forest management”. Martell et al. (1998) also mentions that certain OR models have been included in California legislation regulating the long-term management of forest estates. Martell et al. (1998) lists OR approaches that have been applied to forest operations, fire management, sawmill optimisations, distribution of the forest products to market and, in the area of particular relevance to this thesis, forest planning.

[5]*Pinus radiata* requires 20-30 years to reach maturity in New Zealand.

2.7.1 Forest planning hierarchy

Forestry planning operations have traditionally been divided into a hierarchy of plans. This hierarchy is shown in Figure 2.7, and discussed in detail in Section 4.1.

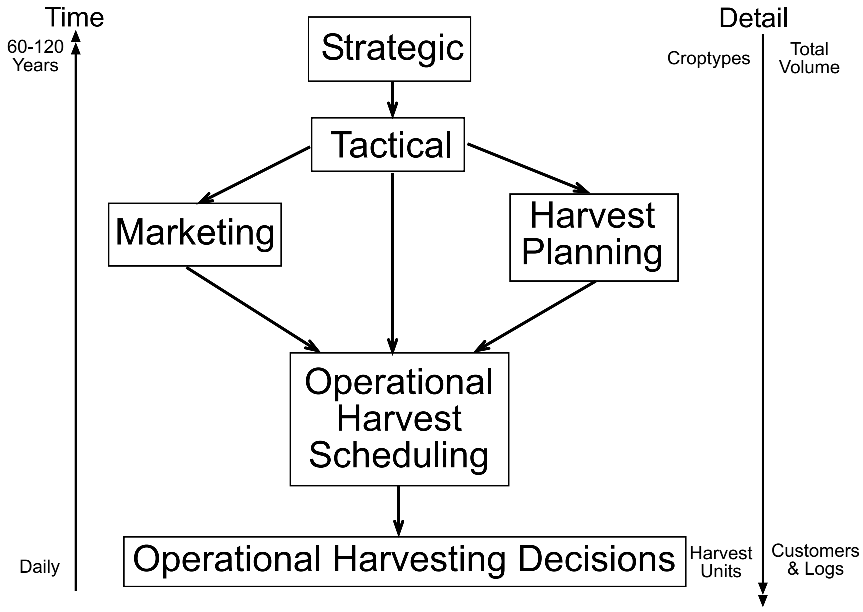


Figure 2.7: The forest planning hierarchy

The *Strategic plan* is the longest term planning model in the hierarchy. With a time horizon of two to four rotations, the strategic plan considers the long-term sustainability of the forest, along with questions of land use and silvicultural regime analysis. The results of the strategic plan indicate aggregate volumes that can be obtained from a forest in the next 1-5 years and the age-classes that should be harvested, without indicating precisely which stands and compartments will contribute to this volume.

The *Tactical plan* has a two to five year planning horizon. The tactical plan uses the results of the strategic plan to determine which actual stands and compartments should be harvested in the next three to six months. The tactical plan can include environmental constraints and the cost of road building when it decides on the actual stands and compartments. Tactical planning is based on aggregate log groupings, and not on the individual log-types for the customers.

The *Marketing plan* directs the sale of the available production of the forest. The marketers should identify the volumes required by the customers in the coming weeks, as well as the sales price, and any flexibility that the customer has. For instance, customers may be willing to accept volume from higher value log-types than they require (downgrades), or perhaps the customers will be satisfied with less volume than agreed.

Harvest planning occurs after the tactical plan has identified the stands and compartments and before they are harvested. As well as determining what kind of operation is required, harvest planning will determine the details of that operation, the required road building and clearing operations and the landing sites within the harvest unit. For a ground-based unit the plan of harvesting will be proposed and for a cable-logging unit the locations and type of machinery are determined. The harvest plan will also decide on the exact boundaries of the harvest units (see Section 3.2) which may be different from the stands or compartments found in the tactical plan.

The *Operational Harvest Scheduling (*OHS*) plan* is the subject of this thesis and is described in depth in Section 3.1. OHS is the last planning operation to take place in production forestry. It occurs weekly and decides the schedule for the next week. The OHS plan is based on decisions that occur in longer time scales in the strategic and tactical plans. Input information is also sourced from marketing and harvest planning decisions. Once the OHS plan for a week has been found, day-to-day operational decisions are based on its results. This is shown diagrammatically in Figure 2.7.

Literature relevant to the strategic, tactical plans and the OHS is discussed in Chapter 4.

Chapter 3

Problem Description

So twice five miles of fertile ground
With walls and towers were girdled round:
(Coleridge 1798)

This chapter describes the physical characteristics of the Operational Harvest Scheduling (OHS) Problem. The problem description is specific to forestry operations in plantation forests in New Zealand and Australia described in Chapter 2, though the general principles can be extended to other countries' industries.

This chapter is divided into five parts. Section 3.1 is the overview of the problem where the decisions made in an OHS are discussed.

Section 3.2 covers the creation of the harvest units, how the choice of harvest units available to the OHS is derived from other forest plans and how the location of the harvest units affects other aspects of the problem.

Section 3.3 describes how yield predictions used in the OHS are not fixed but can be iteratively modified. The ability to create a better OHS by iteratively changing the yield is discussed, as well as how different methods of generating yield predictions can affect this process. There is a discussion of the effects of uncertainty in the yield predictions. The possible interpretation of prices generated by iterative yield generation is also discussed.

Section 3.4 details the effects of harvesting operations on the OHS problem. It discusses the productivity and cost of crews. The costs and penalties associated with moving crews are considered. Once an OHS is created, the crews are instructed to harvest according to that schedule. How the results are translated into instructions is described. These instructions should be easy for the crews to implement in the forest.

Section 3.5 discusses the market forces and demands that drive the OHS. How the customers' demands are communicated to the forestry company is described. The ability to hold logs in inventory to meet future demand and the transportation of logs to the customers is detailed.

3.1 Operational Harvest Scheduling

Other plans in the planning hierarchy (see Section 2.7.1) give a operational harvest scheduler information about harvest units and market demands. The scheduler requires information that details the capacities and costs of the crews and also the yield predictions for the harvest units. The scheduler uses all this information to produce an OHS for the coming week.

The OHS is used in practice to give daily instructions to the forestry crews and to allocate the production from the forest. Any adjustments to the schedule within the week are reported and used to adjust the schedules generated for the following weeks.

3.1.1 Decisions made in the OHS

The OHS determines week by week how the forest will be harvested. The physical description of the harvesting problem is detailed in Sections 3.2–3.5. An OHS solution will have three attributes.

- The location of harvesting crews throughout the time horizon is given, i.e., which harvest units a crew will harvest in each week. The movements and idle periods for the crews are included in this information.
- The crews are instructed to harvest particular log-types and volumes. This production is determined by yield predictions for the harvest units.
- The crews' production is allocated to specific customers. Any shortfall in production to customers is determined. If necessary instructions are given to purchase logs from outside the forest to meet customer demand.

3.1.1.1 Current practice

The OHS problem is currently solved manually in a large number of New Zealand forestry companies. Typically, a general-purpose tool such as a spreadsheet is used along with trial and error by an experienced harvest scheduler. The schedules generated by this method are usually based on the schedule of the preceding week, with changes made only when necessary or when they give obvious benefits.

Often a schedule that meets the various constraints is difficult to find manually. Customers' demands may not be met and significant log-stocks could be accumulated through mismatches of production with demand. More restrictions, or larger schedules, quickly make the problem impossible to solve.

A computational method that solves the OHS problem should produce better quality schedules than those produced manually. Promising results are found in literature. Epstein et al. (1999b) reports a 5-8% increase in net revenues for Chilean forest companies that have implemented an OHS system. In New Zealand, Fletcher Challenge Forestry (FCF) has moved towards an automated method of generating Operational Harvest Schedules for the Central North Island forestry estate (personal communication R. Mills, 2001). The use of this system has led FCF to better regulate the production and allocation of their resources. This system has also allowed FCF to meet the strict requirements necessary for FSC[1] certification. *forest research* is also actively involved in the development of an automated OHS system.

3.2 Harvest units

The OHS chooses harvest units from a set of harvest units that are available in the short-term. This set of harvest units is derived from the results of longer-term planning processes, the Strategic and Tactical plans. The relationship between these plans is discussed in detail in Section 4.1. These plans ensure that all of the harvest units considered by the OHS can be harvested within the OHS planning horizon.

Inventory information is typically collected within stands not within harvest units. This is because harvest planning does not define the boundaries of harvest units before inventory assessment takes place. Therefore, inventory information may need to be adjusted to the bound-

[1] Forest Stewardship Council.

aries of harvest units before it is used in the OHS. In many cases, especially in areas suitable for ground-based operations, stands are identical to the harvest units, but occasionally actual harvest units may be very different from stands.

Clearfelling a harvest unit fulfills two purposes: to harvest the trees in the unit; and to ensure the unit is ready for replanting. Though production and market factors are important in the OHS the operational need to complete a harvest unit must also be considered. A harvest unit must be completely clearfelled before planting takes place. An OHS will ensure that all the harvest units which have been selected for replanting have been clearfelled by the date required in the replanting schedule. A good OHS will tend to clearfell most harvest units in a single operation so that the movement costs for the crews will be minimised. It should not leave a large number of harvest units partly finished because crews move out of the units before they are completed. Any movement of this sort will incur additional costs when the crew returns to finish these harvest units.

Sometimes the order in which harvest units are harvested can become an operational matter. The vegetation in one harvest unit may restrict the access to another. The second harvest unit may therefore be harvested only after the first is completed. Another situation that can occur is when two harvest units are required to be harvested simultaneously, perhaps because the two units are cable-logging units and the set-up costs can be shared between the two units if they are harvested simultaneously.

In general, there are three types of possible restrictions in the OHS

- That harvest area **B** cannot be harvested before harvest area **A** is clearfelled.
- That harvest area **B** cannot be harvested while a crew is harvesting harvest area **A**.
- That harvest area **A** and **B** must be harvested at the same time.

Although it is not explicitly included in the OHS another common and problematic (for mathematical programmers) type of harvest unit sequence constraint are *Green-up constraints*. These constraints state that a harvest unit cannot be harvested if it is adjacent to a recently harvested unit. These type of constraints have been widely studied in the literature (Section 4.1.2). McNaughton (1998) in particular is very thorough in investigating these issues.

A green-up constraint can be effective for up to five years, or possibly longer. Decisions on this timescale are typically considered in the tactical plan (Section 4.1.2), or perhaps in even longer term plans. In the treatment of the OHS problem in this thesis, it is assumed that some longer-term plan will have accounted for green-up constraints. The green-up constraints are implicitly included in the list of candidate harvest units, so the OHS cannot generate a solution that will violate these constraints.

3.2.1 Location of the harvest units

A harvest unit is a unique patch of land within the forest that is determined from the harvest planning process. When harvest planning is complete and the boundaries of the harvest units are defined, the spatial location of the harvest units and their areas can be calculated accurately. Yield predictions can be based on these precise areas.

Once harvested, logs are transported to the customers that require them. Transporting incurs a cost that depends on the distance that the logs are moved. This distance can be calculated from the road distance that trucks will travel between the harvest unit and the customer. These distances can be estimated or found from a map or GIS system. Obviously, the cost of transportation will have an impact on the overall profitability of the operations. A good OHS will try to meet customer demand from the closest harvest unit to minimise transportation costs.

The distances between harvest units can be important when crews move between them. This issue is discussed in detail in Section 3.4.2.

3.3 Yield generation

The yields of a harvest unit are not fixed. This is an important property of the OHS problem. The proportions and types of logs made in any particular harvest unit can change. It is therefore appropriate to find the yield predictions that will give the best solution to the OHS problem.

The accuracy of the solution to an OHS problem depends on the quality of the input information. The accuracy of the yield predictions needs to be considered, if the yield predictions are inaccurate the OHS solution may allocate logs that are impossible to harvest from a harvest unit. In contrast to the usage of yield predictions in longer term planning problems, in the OHS problem only a fraction of each harvest unit is harvested in each period. Therefore, estimates of variation based on the entire area of the harvest unit need to be altered.

3.3.1 Changing yield predictions

The quality of the yield predictions greatly affects the solutions to the OHS problem. The literature on bucking optimisation (Section 4.2.1) and full OHS formulations (Section 4.2.2), tackles this property in two ways. Either by pre-generating a selection of yields, or by iteratively generating yields as the solution strategy progresses. A comparison of these two methods in the literature is found in Section 4.3.

In the discussion below, it is assumed that the yield predictions are generated by a DP bucker using stem descriptions (see Section 2.4.5.2). This method of yield generation is chosen as it will give theoretically optimum results if used in an iterative OHS algorithm (see Section 8.5.4 and Cossens (1996)) .

A distinction must be made between the input prices to the bucker and the *market prices* for logs. While the DP prices can be easily altered, the market prices are fixed at the price a customer is willing to pay. The market price is used when calculating the total profit of an OHS and does not need to be equal to any of the prices used to obtain yield predictions.

3.3.1.1 Dynamic programming

A yield prediction from a DP bucker can be thought of as a single choice or 'snapshot' from all the possible yields that can be obtained from the harvest unit. If a different set of prices is used for the log-types in a DP bucker, a different snapshot will be returned.

To further describe this solution feature, imagine the yield predictions are located in a space where each axis represents the volume of a log-type. Yield predictions form the extreme points on the boundary of a frontier that indicates the best possible (efficient) uses of the harvest unit. The yield predictions represent single solutions from the DP bucker. These solutions are discrete points in this space and do not change continuously with a changing price vector.

Firstly, consider two input price vectors ($\boldsymbol{p}_1, \boldsymbol{p}_2$) representing the prices for each log-type. If one vector is the scalar multiple of the other $\boldsymbol{p}_2 = c\boldsymbol{p}_1$. The yield predictions obtained from these vectors (the optimal solution to the dynamic program) will be identical. As an optimal solution $(\boldsymbol{x}^*)$ in the DP for the first price vector

$$\boldsymbol{x}^* \in \arg\max_{\boldsymbol{x}\in\mathbb{R}^n} \{z_1 = \boldsymbol{p}_1\boldsymbol{x}\} \quad z_1^* = \max_{\boldsymbol{x}\in\mathbb{R}^n} \{z_1 = \boldsymbol{p}_1\boldsymbol{x}\}$$

is also optimal for the second

$$\boldsymbol{x}^* \in \arg\max_{\boldsymbol{x}\in\mathbb{R}^n} \{z_2 = \boldsymbol{p}_2\boldsymbol{x}\} \quad z_2^* = \max_{\boldsymbol{x}\in\mathbb{R}^n} \{z_2 = \boldsymbol{p}_2\boldsymbol{x}\}\,.$$

However, the optimal objective value for each price vector will be scaled so that $z_2^* = cz_1^*$. Therefore, the decisions made in the DP are based on the comparison of the relative prices between competing log-types.

In recognition of this feature, input prices to a DP bucker are known as *relative prices* because the relativity between the log-types is important. If the relativity between log prices is changed, the yield prediction will change in discrete steps. These changes in the yield prediction will only occur when the prices change to make alternative decisions viable within the DP. To illustrate this, consider a yield prediction with two logs-types: a single length of sawlog; and a single length of pulp. A diagram of a possible frontier is illustrated in Figure 3.1.

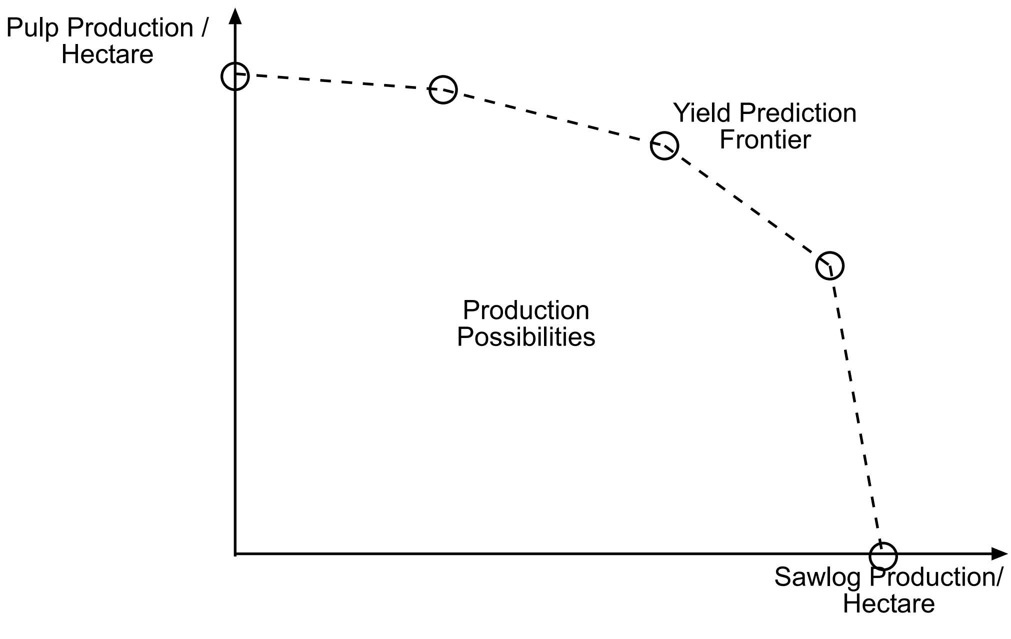


Figure 3.1: The theoretical yield prediction frontier

There is a point on each axis that represents the solution for a relative price vector where one log-type is at zero value and the other is at a positive value. Therefore, the solution is one log-type at zero volume and the other at its maximum obtainable volume. If we follow the frontier from the bottom right to the top left, we observe discrete shifts in the volume by log-type of the yield prediction. When the price vector is changed so that both log-types have a value, but sawlogs are favoured over pulp, the yield prediction abruptly changes. The yield prediction then shows the maximum volume of sawlogs possible with the left over volume converted to pulp if possible. The yield prediction will remain at this point, as the price vector changes, until some new relativity is reached between pulp and sawlog prices. At this stage, another solution becomes optimal. Eventually when the price vector has a zero value for sawlogs and pulp at a positive value the solution maximises pulp volume with no sawlogs produced.

This yield prediction frontier is directly equivalent to the convex hull extreme points produced by a parametric analysis of changing objective function coefficients in an Linear Program (LP). This equivalence can be established by reformulating the bucking problem as a shortest path Network Flow Problem (NFP). The solution to the NFP will be naturally integer and therefore equivalent to the solutions from the DP recursion.

An example of this behaviour taken from Ogweno (1995) is shown in Figure 3.2. The non-linear change in log volumes as a function of price is clearly shown in this example. Note how as the value of log-type 4 is increased from 75 to 155 no change in the solution occurs. When the value is increased over 155, the volume of log-type four increases, while the other log-types begin to decrease in volume.

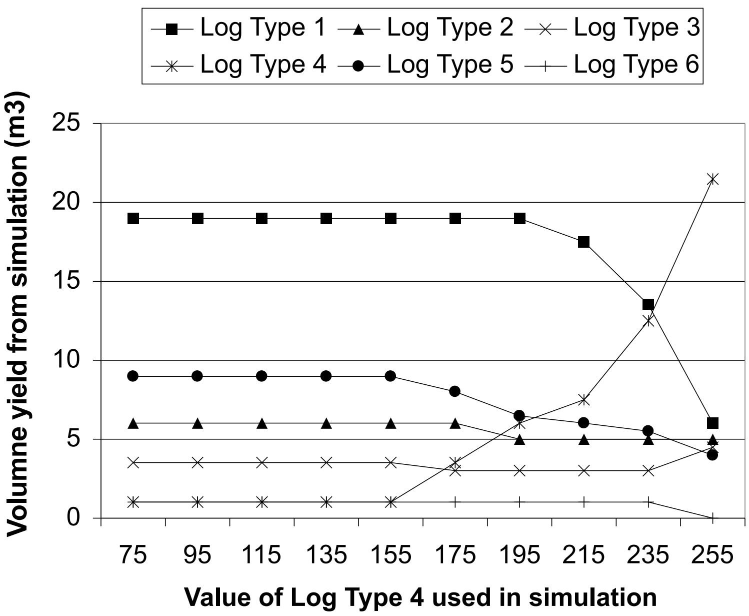


Figure 3.2: Effects of changing the value of one log-type on a yield prediction (Ogweno 1995, pg. 52)

For practical reasons, it can be assumed that the extreme point solutions of the DP can be linked by a piecewise linear convex curve (the lines in Figure 3.1). Though the DP itself cannot find solutions that are continuous, the yields can be assumed continuous in a practical sense. In practice, if harvest units are assumed homogeneous, a harvesting crew could buck using one price vector, then change over to another price vector, giving a linear combination of the outputs from both vectors. The ability to find continuous solutions is required because of the constraints on log volume in the OHS problem.

The complete curve is a frontier that encloses all possible solutions to the bucking problem and all solutions along the curve are efficient. As the DP is guaranteed to produce optimal solutions, all other feasible solutions will lie within the frontier. All yields must lie within this frontier including yields generated by priority lists, log volume conversions or stem-class techniques, such as those found in Eng et al. (1986). These other methods may produce solutions that are on the frontier but since they are not optimal heuristics, they cannot be used to produce extreme points.

3.3.1.2 Priority list

The solutions given by a priority list bucker, as priorities are changed, behave similarly to the solutions to the DP bucker, when prices are changed. That is, the solutions do not vary continuously and instead form a frontier of points in the solution space. The frontier created by a priority list bucker is not necessarily convex as the list can be manipulated in ways that give a lower objective value than a linear combination of two other solutions.

The solution points from the priority bucker also necessarily lie inside or on the frontier given by the solutions from the DP bucker as solutions given by the priority list bucker are feasible solutions considered by the DP bucker. As the DP bucker gives the optimal value solution for any particular set of prices, the solutions given by the priority list bucker must have objective values less than or equal to the DP bucker solutions.

3.3.1.3 Log conversions

Log conversions also provide solutions that lie within the frontier from the DP bucker. However, unlike the DP and priority list methods, the conversions may give continuous solutions.

3.3.2 Uncertainty in yield predictions

As yield predictions are based on sampled information, they are inherently uncertain. If a well-designed inventory system does not have a bias, the precision of predictions can be quantified using statistical analysis. The statistical functions in an inventory system usually report the uncertainty of the estimates by placing confidence intervals around the predictions[2]. As the predictions are typically per hectare estimates of log volume the confidence intervals expressed as per hectare values as well.

Estimates of total standing volume for a stand tend to be very precise. However, individual log-type estimates can be imprecise as certain log-types can be very rare and not represented well in the sample. In addition, as the log volumes are derived from a stem optimisation, the variability between trees and plots can be large. The optimisation will give extreme point solutions based on specific features of a stem. Therefore, stems that are very similar physically (perhaps differing only by location of a defect) may have very different bucking solutions.

The confidence intervals on the estimates, are calculated from the standard deviation (s_s) of the per hectare volume estimate over the entire stand. In the OHS, only a fraction of the stand is harvested in a period. To calculate the precision of the estimate for this smaller fraction of the stand, the standard deviation over the smaller area (s_f) is found by Equation (3.1).

$$s_f = s_s \sqrt{\frac{a_s}{a_f}} \tag{3.1}$$

where:

s_f	is the standard deviation of per hectare estimates of the fraction
s_s	standard deviation of per hectare estimates of the stand
a_s	area of the stand
a_f	area of the fraction.

The derivation of this equation is found in Appendix B. It should be noted that the standard deviation of the per hectare estimates increases as the area harvested decreases.

The variability of the yield predictions though important is not addressed any further in this

[2] In New Zealand, Probable Limits of Error (PLE) is reported which is the confidence interval expressed as a percentage of the mean (Fisher 1995).

thesis.

3.3.3 Analysis of relative prices

Yield generation as discussed above (Section 3.3.1) will also generate a set of prices or priorities for logs. Though the core outputs of the Operational Harvest Scheduling Algorithm (OHSA) are the projected production targets from each harvest unit, the prices or priorities used to generate these targets give important information. These *relative prices* are distinct from market prices used to calculate revenues. The following discussion refers to relative prices from a DP bucker but is also relevant to priority lists.

Relative prices can show the true value of each log-type. Though they are not the true marginal prices, the relative prices show importance of log-types in the current solution, based on the demand. For instance, if the market price of pulp is $5 per cubic metre and an offer of $7 is made for more pulp; is an increase in pulp production economical? If the relative price for pulp is $11 this indicates that the new price is not high enough to justify extra volume, even though there is a market price increase.

Relative prices can alter between harvest units. The differences between harvest units can indicate the changing costs of meeting customer's demand caused by transportation costs. They also indicate geographically where the OHSA would prefer to source logs to meet specific customers.

3.4 Harvesting considerations

The harvesting factors that are directly related to the OHS problem are the capability of crews to operate in different harvest units and their varying productivity in the units they do harvest. The type of crew also affects how the crew moves between harvest units and the associated costs. The output from an OHS will determine the instructions given to crews. How these instructions are presented is also discussed in this section.

3.4.1 Crews' productivity and cost

The productivity of harvesting crews is measured in cubic metres per day (or week). The productivity of the crew can depend on several factors:

- the type of crew;
- the size of the crew;
- the crew composition;
- the harvest unit in which they are operating;
- the time of year.

The most important factors are the type, size and composition of the crew itself. As mentioned previously, cable crews are significantly less productive than ground based crews. Highly mechanized crews are more productive than those that are less mechanized and larger crews tend to have greater productivity than smaller crews. The productivity for cable operations can range from 50m^3 a day for a small crew on thinning to 250m^3 per day for an efficient crew. Ground based operations can range from 50-1000m^3 per day depending on the crew and operation type.

The type of harvest unit that the crew is operating in can influence the productivity. This influence is comprised of several factors.

- The average size of the stems in a harvest unit: larger stems increase productivity.
- The pruning of the trees: lower branches of an unpruned harvest unit can impede crews especially when production thinning.
- The final stockings of the harvest unit: in lower stockings crews can operate more efficiently as the unit is less crowded.
- The slope of the harvest unit: the weather when harvesting and the soil type in the harvest unit all affect the productivity.

The effects of some, or all, of these factors are found in work-study information collected by the individual forestry companies. The method of calculating the productivity of the harvesting crews is heavily dependent on the type of information that the forestry company has at its disposal. A forestry company might be limited to predicting its crews' productivity based on past performance information. Another may use complex formulae, or computer programs considering all the factors mentioned.

Similar to the productivity, the cost of harvesting is determined by the crew and the terrain. The harvesting crew will usually contract to harvest an area on a $\$/\mathrm{m}^3$ basis. There have been some initiatives to provide payment incentives based on value recovered, but the most common arrangement is to simply pay by volume.

3.4.2 Movement and set-up

When a crew finishes a harvest unit, it will need to move to another harvest unit. When moving, a crew will stop harvesting, clear its equipment from the old site and set up the equipment in another harvest unit. This process takes time and may incur costs if trucks or other methods of transport need to be brought in to move the crew and machinery. The time taken and cost incurred obviously depend on the type and composition of the crew. A ground-based crew will be able to move faster with less cost then a cable logging operation. In fact, the costs of moving a cable logging operation and the time lost make it prohibitively expensive to move a cable crew before a harvest unit is completed.

Finishing the harvest unit is also advisable for a ground based harvesting crew however, these crews can be moved early if there is an urgent demand elsewhere. If the crew does leave a harvest unit unfinished, at some point a new crew will have to re-enter the harvest unit and finish harvesting so that the area can be replanted. Leaving unfinished harvest units can therefore affect the long-term profitability of operational harvesting.

The costs and time loss of movement have a fixed component that reflects the costs mentioned above. There also is variable component that reflects the distance moved. If the distance moved is large, the time component of actually travelling can become significant; also the costs per kilometre to transport the machinery can become important. In many cases the variable costs and time penalty for the transportation will be small compared to the fixed costs of moving. Often these distances only need to be taken into account if they are large e.g., when a crew moves between two forests.

Some types of crews (particularly ground-based) are able to move at any time during the week. Cable logging operations however, may prefer to move and set up in a new harvest unit during the weekend. Moving during the weekend is preferable as the entire crew is not idle while the equipment is moved. Instead, only the members of the crew necessary for the moving operations will work.

3.4.3 Instructions to crews

Instructions are given to the harvesting crews that determine, both where the crews will operate and what they should produce in the coming week. These instructions are interpreted by log makers when they decide how the stems are bucked into the logs. Typically, these instructions are a priority list that gives a clear indication to the log maker on how to decide which log-types are made from the stem. If the crews are using a system similar to IFRLogger (see Section 2.5.1.4) then each log-type on the list is associated with a relative price (often different from market prices). The log making system attempts to optimise the value of each stem that is examined, based on the relative prices. Another alternative is to instruct the crew to produce the volumes required, and to allow them to adjust the prices and priorities to achieve these targets.

3.5 Market considerations

This section discusses the market considerations that affect the OHS problem. This includes the representation of demand requirements, transportation of logs, the accumulation of log-stocks and the ability to transfer volume from one log-type to another by downgrading.

3.5.1 Demands

In the short-term, there are two different demand analyses.

- Capacity planning and marketing analysis occurs when the customers' orders are confirmed and the target volumes are set.
- Operational allocation of the actual production to the customers occurs during production.

Recurring demand (over several periods) could be in several different forms, depending on the customers' requirements. A pulp mill may require a constant volume of pulpwood every week to keep it running at its optimal efficiency. If the pulp mill keeps a large stockpile of logs, the actual volume delivered each week becomes less important as inventory at the pulp mill will handle any fluctuations. If the pulp mill wishes to reduce the cost of holding inventory however, the weekly target volume becomes more important.

Sawlog customers may require changing volumes of different log-types depending on the timber products the sawmill is producing in the week. Again, if the sawmill keeps inventory, the target amounts could be flexible.

Export logs are delivered to a port and placed on ships for export. These ships require a large volume of wood to fill them. As the ships are very expensive to keep in port, the logs are loaded onto the ship as quickly as possible. In order to quickly load logs onto the ship, logs are harvested in advance and stockpiled at or near the port before the ship arrives. Therefore, this volume of sawlogs has to be ready and waiting, before the ship arrives at port. Weekly targets are not essential, only the total volume required before the ship docks.

Demands may also be expressed as the sum of the delivered volumes of several log-types. Individual log-types within this sum may be restricted by volume, fraction of total volume, or log properties such as SED. Examples of these aggregate requirements are given in Section 2.6.3.

Ideally, the planned demand targets are achievable and met by the production in each period. However, if the demand targets cannot be met by production, there must be some way to allocate the log-types available to the required customers. The demand and production may not match up for several reasons. The log marketers may have over sold log-types and the forest and crews are not capable of achieving the volumes necessary in the log-types required. The forest resource may significantly differ from the estimates used (i.e., the forest cannot produce the volume of higher quality log-types required). Alternatively, an unforeseen event may have reduced the production in that period.

Regardless of the reason for the shortfall, the customer demands must be taken into account and logs can be supplied from external sources, or some customers may go unsatisfied. The decisions made will take into account the relationships with the customers, any financial penalties for not meeting contracted volumes and the cost of buying logs from a third party.

3.5.2 Log-stocks

Once stems have been bucked, the logs are placed in piles on the landing until they are delivered. These stockpiles are often referred to as log-stocks. They are usually not called inventory as this can cause confusion with the usage of inventory to refer to *standing inventory* (see Section 2.4).

The total volume of the stockpiled logs across the forest estate can be quite significant and therefore represents a large capital expense to the company. Logs placed in log-stocks can degrade significantly if left for long periods. A common form of log degradation is caused by sap stain, a fungal infection of the wood that discolours the logs and renders them unsuitable for appearance grade products.

The financial cost of keeping log-stocks and the associated degradation in the quality of the log-types are important motives for reducing the volume of the stocks. Log-stocks tend to accumulate when logs are made at the landing and then not sold to customers. These logs will then remain at the landing until a customer is found or the logs are degraded into some other log-type that a customer demands. Cutting logs that precisely fit the customer demand can achieve reduction in the volume of the log-stocks as the logs are delivered to the customers quickly and not left to degrade on the landing.

As the landing sites only have a limited area to store logs, it is not possible to store large amounts of logs at a single landing.

3.5.3 Transportation of logs

Logs are delivered to the customers, who can be quite distant from the forest itself. To minimise double handling costs, the logs are commonly taken directly from the landings (or super skids, or central processing yard) where they have been bucked. As the landings in the forest are likely to be scattered geographically and the price of transporting logs can be large (approximately \$15-\$20 per 100km per m^3 (M.A.F. 2002)) it is sensible to meet customers' demands from the closest site possible. In some cases, this may not be practical because the closest harvest unit does not contain resource of sufficient quality, or the total volume required by the customer must come from several harvest units.

In some forestry companies, logs are sold on a stumpage basis (see Section 2.6.4) and the price of transportation is passed on to the customer. The company may even place a margin on the transportation cost and thus actually profit if the logs are sourced from harvest units that are distant from the customer. Though an accounting perspective would indicate the desirability of supplying customers from these distant units, customers would not be pleased with this solution and if the company were operating in a competitive environment the customer would soon change supplier.

3.5.4 Downgrading

A downgraded log meets the specifications of a high quality log-type, but is instead sold as a lower quality log-type. There are two ways to downgrade logs in the harvest units.

- A high quality region of the stem could be selected by the log maker to make low quality log-types
- The already bucked logs on the landing can be substituted for lower quality log-types.

The degrading of stems by the log maker can be simulated by the inventory package (see Section 2.4). The inventory system can allocate lower quality logs in preference to the higher quality logs with the manipulation of the relative prices given to the inventory system.

Downgrading of already bucked logs can be achieved either by selling the higher quality logs as the other log-types directly or, perhaps, trimming old logs to meet the specifications of other products. For instance, if the required log-types have a lower large end diameter specification then the logs on the landing, the logs will need to have their large ends removed so that they are suitable for the new log-type. In this case, the volume of logs degraded will not be equal to the final volume of the logs sold, as the residual parts of the log will be wasted.

Downgrading of logs, after they are bucked, can be undesirable. Re-grading by generating new yield predictions can more accurately model the processes that occur operationally. However, if a large volume of log-stocks is available at the beginning of the planning horizon, downgrading may be necessary to eliminate these large stockpiles. This was the case in one of the case studies (see Section 9.5). In this case study, it was beneficial to limit downgrading to these initial log-stocks. Newly harvested logs were only re-graded through the yield generation process.

Chapter 4

Literature Review

And there were gardens bright with sinuous rills,
Where blossomed many an incense-bearing tree;
(Coleridge 1798)

This chapter reviews hierarchical planning systems applied to forestry, and describes the purpose and benefits of hierarchical planning found in forestry literature. The purpose and place of strategic and tactical forest planning systems are discussed and examples of these systems from literature are given.

The literature on Operational Harvest Scheduling (OHS) is reviewed in detail. This literature includes work on *Bucking optimisation* techniques that solve a restricted form of the OHS problem. The problems solved in the literature, the techniques used to solve them and any results and conclusions are discussed.

4.1 Hierarchical forest planning

It is common practice to divide forest-planning operations into a hierarchy of strategic, tactical and operational plans. The levels in *hierarchical planning* are described in Martell et al. (1998, sections 3.1-3.3), though the authors use the term *intermediate range planning* to describe what is referred to as the tactical plan in this thesis.

An earlier paper, Gunn (1991), describes the hierarchical planning paradigm in depth. The author includes a table of definitions for these plans which is shown in Table 4.1.

Table 4.1: Characteristics of decision problems in hierarchy (Gunn 1991)

Characteristics	**Strategic Planning**	**Tactical Planning**	**Operational Control**
Objective	Resource	Resource acquisition	Execution utilization
Time Horizon	Long	Middle	Short
Level of Management	Top	Middle	Low
Scope	Broad	Medium	Narrow
Source of Information	External & Internal	External & Internal	Internal
Level of Detail	Highly Aggregate	Moderately Aggregate	Very Detailed
Degree of Uncertainty	High	Moderate	Low
Degree of Risk	High	Moderate	Low

An interesting feature of hierarchical planning is that all plans include a single common starting point, the current period. The plans differ in resolution and accuracy of this first period and the number and length of the other periods in the planning horizon.

While describing hierarchical planning, Martell et al. (1998) comments on the difficulty of dealing with such a large amount of data and the long time horizons of the plans.

> *Foresters and their OR counterparts have therefore been forced to a greater degree than those working in many other areas, to address questions concerning the appropriate level of detail in individual models and decision making processes at various levels in the hierarchy.*

Later the authors include this comment.

> *Clearly, there is enough here to push hierarchical planning methodologies to their limits.*

The question of transferring decisions and constraints between models within the hierarchy is discussed in Laroze & Greber (1991) and in Ogweno (1995). In both, the authors describe integrated hierarchical planning systems, which encompass strategic, tactical and operational components. The strategic and tactical plans are formed as linear programming models that differ in the level of detail considered and also in the time horizon.

Ogweno (1995) presents a hierarchical planning system that includes strategic, tactical, operational and yield prediction components. The strategic planning tool used is FOLPI (García 1984). The tactical planning tool used is also FOLPI, although the data and constraints are altered from the strategic plan. The OHS system and the stem bucking optimiser (XCut) were developed by the author.

In Epstein et al. (1999a), several OR systems that have been developed by researchers at the University of Chile are described. The systems are listed below.

- ASICAM: a truck scheduling tool.
- OPTICORT: an operational harvest scheduling system.
- PLANEX: a machine-location and road design system.
- OPTIMED: a tactical planning tool.
- MEDFOR: a strategic planning tool.

The implementation of these systems was supported by both the Chilean forestry industry and also a government funding body (Fondef). Epstein et al. (1999a) stresses the benefit of these tools to industry.

> *..., due to the use of three OR systems, Bosques Arauco reports total savings of (US) $8 million a year over a total annual timber production worth $140 million. The firm considers its use of these systems a strategic competitive advantage.*

Table 4.2 summarises the general characteristics of each problem in the forestry planning hierarchy.

4.1.1 Strategic planning

It must be noted here that according to Gunn (1991) strategic decisions define the role and nature of the organisation and the resources that the organisation will have available. These decisions are made by top-level management of an organisation using a combination of experience, simulation, and guesswork. In forestry, however, it is common to use the term *strategic planning*

Table 4.2: Characteristics of forestry planning problems

Characteristics	**Strategic Planning**	**Tactical Planning**	**Operational Control**
Time Horizon	2-3 rotations	5-10 years	2-6 months
Time Periods	Years	Years /Months	Weeks
Decisions	Silviculture Harvesting Land use	Harvesting Road Building Environmental Product usage	Harvesting Product Allocation
Area unit	Croptype	Stand /Compartment	Harvest Unit
Yields	Predicted	Predicted Cruised	Cruised
Number of Yields	One	One	Many

(or estate modelling) to refer to the simulation or optimisation of long-term (2-3 rotation) forest management decisions. These models, according to strict nomenclature, actually make tactical decisions according to the definition in Gunn (1991). They are used to simulate the effects of strategic decisions (i.e., resource allocation and capacity expansion) on the profitability of a well-run forest.

The usual formulation of a strategic plan is to determine the management regime and the times of harvest of areas within the forest. The areas used in strategic planning are *Crop-types*. Crop-types are aggregate areas of the forest of similar silviculture and yield, each crop-type being further divided into *Age-classes*, which are used to determine the maturity of the crop. The results from the strategic plan determine the area, by crop-type and age class that is harvested in each planning period.

The strategic planning problem is commonly formulated as an LP, though Martell et al. (1998) does mention there is still some debate.

> *...despite the prominence of linear programming based models, debate continues in the forestry community comparing them to detailed simulation models with operations researchers often coming down on the side of linear programming and forest mensuration specialists preferring simulation approaches.*

Other optimisation approaches apart from LP have been attempted and a good list is found in Barros & Weintraub (1982, and references within) but the authors do draw this conclusion.

However, none of these (approaches) has been implemented regularly. This is because of the excessive simplification of the real life situation required by the models, or due to difficulties in implementation.

The standard formulations of strategic LP formulations are described in Johnson & Scheurman (1977). Following the nomenclature of Johnson & Scheurman (1977), it has been common for subsequent authors to describe their models as either a Model 1 or Model 2 formulation. These models differ in the relationship between decisions on management and harvesting, and particular crop-types and age classes. Where a Model 1 formulation preserves the identity of a particular crop-type and age class throughout the time horizon, a Model 2 formulation can aggregate them at the time of harvest. García (1990) presents a *revised classification* of the models presented in Johnson & Scheurman (1977), by describing three different classifications (A, B and C) that use and clarify the concepts of model formulation.

Examples of stand-alone strategic planning tools are listed below.

- FORPLAN: described in Johnson et al. (1986).
- Spectrum: described in USDA Forest Service (1995).
- MEDFOR: described in Epstein et al. (1999a).
- FOLPI: described in García (1984).

Manley & Threadgill (1991) presents a case study of the use of strategic planning to give a valuation of a large proportion of the New Zealand forestry estate that was owned by the government (and since privatised). The use of a strategic planning tool is described by the authors as necessary to determine a value for the forests based on discounted expected net cash flows. This paper quotes from a senior government bureaucrat at the time of the asset sales.

No attempt has been made to determine the monetary benefit of the FOLPI system in the work reported. ... It would have been difficult, if not impossible, to complete the investigations without such an analytical tool.

Within this paper, the authors describe an extension to the traditional strategic planning model that includes log allocation constraints.

4.1.1.1 Long-term log allocation

Log allocation constraints model the market for forestry products. Traditionally the market model has not been part of the strategic plan, as it is assumed that the market capacity can be altered on smaller time scales (3-5 years) than the productive capacity of a forest (30+ years). These constraints are often modelled implicitly in strategic formulations with non-declining yield or other constraints. García (1990) provides a detailed description of these constraints named “The Utilisation Submodel” and their various uses. The author contends that it is sometimes more appropriate to use these implicit constraints instead of explicit market models.

> *…the development effort and the information requirements can be considerable. In addition, often the problem details are not sufficiently well defined, as in many indicative planning studies.*

Sometimes it is necessary to include market constraints within a model especially for the first few periods. The inclusion of a log allocation model in FOLPI is described by Manley & Threadgill (1987). An integrated approach to strategic planning is described in Barros & Weintraub (1982), where decisions on the usage of timber (pulp, sawlogs or export logs) are included in the strategic planning model.

McGuigan (1984) is an interesting case study of LOGRAM-1 a log allocation model used to determine the allocation of logs within a conglomerate New Zealand company. This model did not include the forestry decisions found in strategic plans, as the forest data came from a simulation model IFS (also described in García (1990)). The log allocation model consists of a simple transportation model that would meet demand while minimising cost. This approach is similar to Goulding (1974) a very early paper, where a simple log allocation transportation problem is formulated on an ICL 1904 computer. This report includes an interesting quote that is applicable to all optimisation models.

> *Finally, the discipline of having to determine all the factors and constraints that affect the distribution problem and arrange them in a logical order may be of as much benefit to the forest manager as the solution to the problem.*

4.1.2 Tactical planning

Tactical planning deals with medium-term (5-10 years) and resolves the spatial location of the harvest areas. While in strategic plans the land unit is a crop-type, in tactical plans *stands* are used. A stand has a unique spatial location.

One approach to solving tactical problems is to simply dis-aggregate the large crop-types in the strategic plan into many individual stands then solve the problem accordingly. Examples of this approach are found in Papps & Manley (1992) , Laroze & Greber (1991) and Ogweno (1995). In Papps & Manley (1992) a change in the FOLPI Model (García 1984) to include variable period and age class lengths is discussed as is the ability to aggregate the stands to be harvested in the short term into crop-types when they are replanted.

An important aspect of tactical planning is the inclusion of spatial constraints. These constraints model physical constraints of harvesting or environmental issues. The nature of these constraints will force a stand or compartment to be harvested or left unharvested depending on whether other stands in its neighbourhood have been harvested previously.

Spatial constraints based on harvesting requirements can be due to road building considerations or cable harvesting requirements. These constraints are inclusive, forcing a stand to be harvested when the stands close to it have been harvested. Road building constraints consider that roads are built to access stands before they are harvested. Once a road has been built to access an area, it is economical to harvest other stands accessible by the road at the same time. This issue is addressed in detail in McNaughton (1998, section 3.8).

Literature has focused on the environmental constraints contained in tactical planning. These constraints are very important in North American forestry in particular and are known as *Green-up constraints*. A Green-up constraint limits the maximum sized clear-cut area permissible in a forest. One form of a green-up constraint restricts harvesting in stands adjacent to a harvested stand. A common example of a green-up constraint is the Forest Stewardship Council (FSC) North American '3.1' standard, where a stand cannot be harvested until its neighbours have reached a height of 3.1 metres. There are many methods of formulating this constraint, as the definition of adjacency, area, and regrowth can be altered.

A description of the different types of adjacency constraints can be found in Murray (1999). Green-up constraints are exclusive adjacency constraints that will work against the inclusive adjacency constraints, concerning road building and cable harvesting, mentioned previously.

Adjacency constraints significantly increase the difficulty of finding a solution to the tactical plan. Unlike the strategic plan, McNaughton (1998, section 2.3) comments that there is no one method that has been generally accepted for solving tactical plans.

> *What is curious though, is the wide diversity of methods being advocated even within established research groups.*

From this wide range of methods, the author draws the following conclusion.

> *... not one of the many and varied solution methods has won any degree of widespread acceptance. This is a strong indicator that FHP (Forest Harvest Problem[1]) is at present a challenging unsolved problem. It is certainly true that FHP is at present unsolved in an optimisation sense.*

McNaughton (1998) contains a thorough review of work relevant to tactical planning and divides the literature into three streams depending on the solution methods used.

- Literature that uses a LP/MILP solution process.
- Literature that uses heuristic (non-optimal) solution processes.
- Literature with multi-stage solution processes, which use more than one solution technique at different stages.

Table 4.3 is a summary of the classifications found in McNaughton (1998, pg. 24-30).

Table 4.3: The different solution processes for tactical planning problems

LP/MILP	Heuristic	Multi-Stage
Weintraub & Navon (1976)	Nelson et al. (1988)	Weintraub & Cholaky (1991)
Kirby et al. (1980)	O'Hara et al. (1989)	Weintraub et al. (1994)
Nelson & Brodie (1990)	Yoshimoto et al. (1994)	Borges et al. (1999)
Papps & Manley (1992)	Murray & Church (1995*b*)	
Murray & Church (1995*a*)	Weintraub et al. (1995)	
McNaughton (1998)	Van Deusen (1999)	
McDill & Braze (2001)	Boston & Bettinger (1999)	
	Bettinger et al. (1999)	

[1]This is the tactical planning problem.

Unfortunately, adjacency constraints, though formed as a means of protecting the environment or recognising opinions of good landscape design, are often considered as an end to themselves in the literature. Thus, the literature devotes much effort to find solutions to tactical plans that do not violate these adjacency constraints, without regard to the actual outcomes. This situation is commented on in Martell et al. (1998).

> *OR specialists then focussed on the computational aspects of adjacency constraints but the extent to which the solutions to their mixed integer programming problem that include such constraints actually contribute to the preservation of natural ecosystem processes is not clear.*

Martell et al. (1998) continues by reviewing literature which to some extent deals with this issue either by looking beyond the adjacency constraints in Bevers et al. (1997), or by using Geographical Information System (GIS) to model these issues in Davis & Barrett (1993).

4.2 Operational Harvest Scheduling

The *Operational Harvest Scheduling* problem is concerned with decisions made in the very short-term (6 weeks to 1 year) within a forest. The decisions detail:

- when harvest units are harvested;
- who harvests them;
- what log-types are made;
- which customers are supplied.

A detailed description of the problem has been given in Chapters 2 and 3. Cossens (1992) presents a good overview of the issues involved in short-term planning in New Zealand.

There is not a wide range of literature (only 13 papers are included here) discussing the OHS problem and there appears to be no common set of concepts in the relevant literature. For this reason, many authors e.g., Weintraub et al. (1993), Murphy (1998) and Ogweno (1995) describe the problem in detail before they describe their solution strategy.

Most literature surveyed describes the basic unit of a forest as a ‘stand’. In the following discussion stand is also used to avoid confusion. However, in the body of this thesis stand is replaced by *harvest unit* as in my opinion this is a more precise term. The difference between these two definitions is discussed in Section 2.3.

The formulations given in the following section will have maximise objective functions unless otherwise stated

4.2.1 Bucking optimisation

Bucking optimisation is a special case of the OHS problem that is discussed in the literature. Bucking is the process of cutting a tree stem into merchantable logs. Therefore, the bucking optimisation problem selects (or generates) yield predictions (Section 2.4) for stands to meet market restrictions. This problem is a simplification of the full OHS problem as it does not consider crew constraints and restrictions.

The need for bucking optimisation work was driven by the realisation that earlier methods for bucking stems optimised the value of each stem and did not consider the overall demand for logs. In Eng et al. (1986), the following comment is made.

> *A stem by stem optimization may thus result in a serious mismatch of volumes of logs supplied and end-use product requirements, thereby reducing the value derived from harvesting the forest resource.*

Two definitions used in this section need to be clarified.

- *Bucking pattern*: Is the specific sequence of cuts that make logs out of a stem.
- *Cutting strategy*: A method of determining bucking patterns when applied to a specific stem. A list of log-types, a set of relative prices and a DP bucker is an example of a valid cutting strategy.

4.2.1.1 Early work, Mendoza & Bare, Eng et al.

Mendoza & Bare (1986) and Eng et al. (1986) contributed early work. In both these papers, the authors independently implement similar iterative methods that use an LP with a *yield prediction subproblem*. In an iteration, the LP attempts to satisfy the demand constraints with the existing yield predictions. The dual variables from the LP are then used to direct the subproblem to generate new yield predictions. The subproblem is solved by a DP in Eng et al. (1986), or modified knapsack algorithm in Mendoza & Bare (1986).

In Mendoza & Bare (1986) and Eng et al. (1986), the yield predictions are *Stem-class* based. A stem class is a grouping of identical stems that are not necessarily found in the same stand. Thus, a specific bucking pattern is developed for each stem-class in the input data. The usefulness of a stem-class based result is questionable, as it requires the harvesting crews to classify each of the stems harvested then use a specific bucking pattern on each. However, the iterative method described in these papers is used in other papers, (Laroze & Greber (1993) and Laroze (1999)) as a comparison. This is because the stem class based method provides an optimal solution to the bucking optimisation problem.

A major difference between Eng et al. (1986) and Mendoza & Bare (1986) is the construction of the market model. Eng et al. (1986) models the log market in the LP and requires log prices and demands. The relevant demand constraint is shown in Equation (4.1),

$$\sum_{i}\sum_{j} a_{ijk}x_{ij} \ (\leq, =, \geq) \ b_k \quad \ldots \quad \forall_k \tag{4.1}$$

where:

i	indexes bucking patterns;
j	indexes stem classes;
k	indexes log-types;
x_{ij}	is the number of stems of class j bucked by pattern i;
a_{ijk}	is the associated volume of log-type k;
b_k	is the required demand of log-type k.

Other literature in this area including Sessions et al. (1989), Cossens (1996), and Laroze (1999) has taken a similar approach, and the market model is similar to the one described in Section 3.5.

Mendoza & Bare (1986) on the other hand models the demand for wood products. This market constraint is shown in Equation (4.2).

$$\sum_i \sum_j a_{ijk} x_{ij} - \sum_p y_{kp} = 0 \quad \ldots \quad \forall_k \tag{4.2}$$

where:

p	indexes processing plants;
y_{kp}	is the volume of log-type k allocated to processing plant p.

In this formulation wood processing plants are modelled, the products made at these plants (not shown in Equation (4.2)) appear in the objective function of the LP while the individual logs (y_{kp}) do not.

4.2.1.2 Sessions et al.

Sessions et al. (1989) describes a system that iteratively adjusts the relative prices of logs until a cutting strategy is found that produces the required target volumes from a single stand. Sessions et al. (1989) solves the yield prediction sub problem by a shortest path algorithm. The use of the same algorithm in the field is also discussed to ensure actual production will meet the projected targets.

In Sessions et al. (1989), stems are not stratified into stem-classes before the solution process begins. Therefore, the entire stand uses a single cutting strategy, instead of stem-class based bucking patterns. The authors justify this change in concepts as follows.

> *We seek a solution that can be readily implemented in the field regardless of stand complexity.*

Specifically, the paper uses a binary search to find a price multiplier, for longer (over 24 foot) logs, that ensures the production of long logs. In an unrestricted solution, short logs are favoured because the Scribner scaling rule (Dilworth & Bell 1984) is used. *Log-scaling* is the method used to measure log volume, and therefore price. The Scribner rule calculates log volume based on a cylinder with the log's small end diameter. Therefore, the total volume (and revenue) from two short logs exceeds the volume of a single log cut from the same stem.

To obtain 80% of the harvested volume in longer lengths, the price of the long logs is increased in the sub-problem. The answer found by the heuristic presented compares favourably with the optimal solution from an integer programming formulation ($715 in the heuristic, $716 in the IP).

However, Sessions et al. (1989) does not address the possible failure of the system if the demand restriction could not be met with a single cutting strategy. The behaviour of DP yield predictions under relative price changes is discussed in detail in Section 2.4.5 and in various literature (Ogweno (1995),Cossens (1996)). This discussion shows that the yield of any single log will change discontinuously as the relative price changes. Therefore, it may be impossible to find a single price that produces a result near to the required figure. This discontinuity is not a problem in the example given in Sessions et al. (1989) as the alternate logs differ in length, and no two logs can be substituted exactly for each other.

4.2.1.3 Laroze and Gerber

Laroze & Greber (1993) describes a method that generates priority list bucking instructions (see Section 2.4.5.3) for a stand, instead of bucking patterns for a stem class as in Eng et al. (1986), or prices as in Sessions et al. (1989). Laroze & Greber (1993) contends that the bucking patterns given in Eng et al. (1986) are not practical.

> *Another drawback of this method is its lack of a pre-specified action when the stem's actual taper, grade or breakage does not allow the realisation of the expected log-type.*

The authors also comment that a priority list method will work better in practice because the cutting strategy approach in Sessions et al. (1989), based on an optimal bucker, requires detailed measurements of each tree.

> *…when applied to smaller trees in an intensive production framework the gains in revenue may not compensate for the increase in production costs implied by this method.*

In Laroze & Greber (1993), a method that uses *Monte-Carlo simulation* (Rubinstein 1981) is applied to a bucking optimisation problem. The problem includes 23 stands of *Pinus radiata* in Chile, and 8 different sets of market constraints (relating to the proportions and specifications of export logs) for 6 log-types (long, intermediate and short export logs, two domestic sawlogs and domestic pulp logs). The method described individually maximises profit for *each* stand while requiring that *each* stand meets export requirements. These export requirements included minimum SED for the stand and restrictions on length proportions.

Laroze & Greber (1993) describes three different Monte-Carlo simulation methods that assign priority lists to the stands. Priority lists generated in the simulation alter the following specifications for export logs:

- minimum end-diameter;
- maximum number of logs from a single stem;
- quality classes allowed.

Domestic log specifications were not altered, nor were the position of log-types in the priority list.

The results from the simulation compared favourably (within 3.5% on average) with the solution of an integer programming shortest path formulation similar to Eng et al. (1986). An integer programming formulation (where x_{ij} is integer (Equation (4.1)) was used as Laroze & Greber (1993) reasons that a single bucking pattern for each stem class will

> *. . . represent the likely outcome to be obtained from a mechanized harvester enhanced with optimal bucking features, or from optimal-bucking machinery implemented with a scanner in a sort-yard.*

Laroze & Greber (1993) compares the reduction in profit between the LP and the IP formulation and finds a difference of 1.07% or less in all cases. The similarity of the LP and IP solution values is attributed to the structure of the problem. The authors also note that the column generation procedure used does not necessarily generate all the bucking patterns needed for an optimal IP solution.

The IP solution is compared to the simulation and the following statement is made.

> *The MC (Monte-Carlo) bucking pattern is characterized by its consistency across diameter and quality classes: in fact, it is possible to derive the underlying rules from the pattern. The IP bucking pattern can be characterized by its irregular behaviour but also by its ability to take advantage of the JAS (Japanese Agricultural Standards) cubic-volume log scale and its sensitivity to profit differentials due to quality.*

In conclusion, Laroze & Greber (1993) describes the success of the system and states that it has been implemented, but also note that the Monte-Carlo simulation may not be sophisticated enough to handle more complicated market constraints. This paper however, fails to mention that the position of the log-types on the priority list could become increasingly important with an increasing number of log-types, as seen in Epstein et al. (1999b). The system described does not change these priorities. As Laroze & Greber (1993) presents a stand-based solution, there is also no ability for the shortfalls of a particular stand to balance excesses of another.

In Laroze & Greber (1997), the authors revisit their earlier case study (though with slight differences in the figures). This time a *Tabu search* (Glover & Laguna 1993) is used to alter the priority list for each stand. The properties altered are the same as in Laroze & Greber (1993). The performance of the Tabu search is an improvement on the Monte-Carlo simulation. The average difference in solution value between the IP and the Tabu search is 2.4% compared with a 3.5% difference described in Laroze & Greber (1993). Laroze & Greber (1997) makes similar comments to Laroze & Greber (1993) when the bucking patterns generated by the Tabu search and the IP solution are contrasted. The similarity may indicate that the differences are caused by the use of the priority list bucking process rather than the method used to generate these priority lists. However, the Tabu search still optimises each stand independently.

In Laroze (1999), a model is developed that solves the forest level bucking optimisation problem. In contrast to the author's previous work (Laroze & Greber 1993, Laroze & Greber 1997) this paper considers the total production of all the stands in the problem to meet the market constraints. The Tabu search method in Laroze & Greber (1997) is used to generate the priority lists that give the yield predictions. In the introduction to Laroze (1999) the author acknowledges that operational limitations (to do with crew allocation), that limit the area harvested at any one time are not considered. If these limitations were considered, fewer stands would be available for harvesting at any one time.

The example problem is the same one found in Laroze & Greber (1993) and Laroze & Greber (1997). Again, a method based on Eng et al. (1986) is used for comparison. However, a linear programming approach is used, rather than the integer program seen in Laroze & Greber (1993) or Laroze & Greber (1997). The formulation is altered to include multiple stands, which requires the additional constraint shown in Equation (4.3) (note, the indices have been changed to remain consistent).

$$\sum_j x_{sij} \leq N_{sj} \quad \ldots \quad \forall_{sj} \tag{4.3}$$

where:

s	indexes stands;
N_{si}	is the number of stems in stand s, class i;
x_{ij}	is the number of stems of class j bucked by pattern i.

The LP in the Linear Programming/ Tabu Search (LP/TS) method in Laroze (1999) is similar to the LP derived from Eng et al. (1986) but is area based instead of stem based.

$$\begin{aligned} \sum_{sp} \upsilon_{sp(m)} y_{ip} &\geq V_m \quad \ldots \quad \forall_m \\ \sum_{p} y_{sp} &\leq S_s \quad \ldots \quad \forall_s \end{aligned} \tag{4.4}$$

where:

p	indexes bucking rules;
m	indexes markets;
y_{sp}	is the area of stand s bucked with rule p;
$\upsilon_{sp(m)}$	is the volume produced for market m in stand s using bucking rule p per unit area;
V_m	is the total volume required for market m;
S_s	is the area of stand s.

In contrast with the method in Eng et al. (1986), the Linear Programming/ Tabu Search (LP/TS) method presented in Laroze (1999) is not an iterative algorithm. Instead, 11 different demand scenarios (including domestic only and pulp only) are presented to the Tabu search algorithm given in Laroze & Greber (1997), which generates 11 different yield predictions for each stand. From these pre-generated yield predictions, the LP selects the best combinations to satisfy overall market demand. Laroze (1999) presents a number of different tests for the solution method.

- The number of alternative yield predictions is altered.
- The demand restrictions are altered.
- The relative areas of the stands are altered.
- The prices of the logs are altered.

The results of these tests are straightforward and are explained in detail. An increase in the number of yield predictions considered increases the value of the solutions given by the LP, to within 2% of the optimal solution from the LP/SP method. The increase from 3 yield predictions to 11 only improves the objective from 1-1.5% in the examples given.

Laroze (1999) states that there is a tendency to select only a single yield prediction per stand.

> *...as a direct consequence of the problem's nature that does not permit the LP solution to present a larger fragmentation: the number of basic activities cannot exceed the number of constraints.*

The author defines fragmentation as the number of yield predictions used in the solution compared to the number of stands. It is interesting to note, that when the paper analyses the fragmentation of the solutions to the problem, in 10 out of the 13 scenarios considered there were exactly 27 yield predictions in the optimal solution, while 24 yield predictions were used in the case with no specific demand constraints. Though not discussed in Laroze (1999), it can be speculated that the three extra yield predictions are needed for the three constraints in the demand restrictions. The shipment requirements may not require any extra yield predictions possibly because of the nature of the generated yield predictions.

With more complex demand restrictions the number of required yield predictions, and therefore fragmentation, may increase dramatically, especially when the number of available stands is reduced because of crew capacity requirements. Laroze (1999) draws the following conclusion.

> *Consequently, there would not be a significant loss in global profit if a reduced number of bucking alternatives were considered in the LP model, as long as the activities included are stand-level efficient instead of general purpose bucking rules not designed for specific stand conditions.*

This conclusion may only be valid for simple demand restrictions. For more complex demand constraints, the number of yield predictions used in the solution may increase substantially An example of this behaviour is found in Epstein et al. (1999b) where in one solution 51 yield predictions are used for a single stand.

4.2.1.4 Cossens

Cossens (1996) presents a forest wide, multi-period, bucking optimisation model that uses a decomposition method similar to Eng et al. (1986). However, unlike Eng et al. (1986), the yield predictions are not stem based. The yield predictions are generated by Method of Assessment of Recoverable Volume by Log-type (MARVL) (Deadman & Goulding 1979), a New Zealand inventory system that uses a DP bucking algorithm.

The method and model presented in Cossens (1996) is similar to that in Laroze (1999) with two important differences; the method is iterative, and a DP yield prediction is used. Importantly, Cossens (1996) also presents a method for calculating the upper limit on the solution value.

The formulation described in Cossens (1996) is very general and does not give the exact constraints that are implemented. The problem is formulated as a *Dantzig-Wolfe decomposition* (Dantzig & Wolfe 1960). The restricted master problem (RMP) models the resource and demand constraints, and the sub-problems are the generation of the yield predictions for each stand.

The yield predictions generated by MARVL are obtained from a cutting strategy with a set of transfer prices (or relative prices) generated from the dual variables of the Restricted Master Problem (RMP). If the RMP has includes the volume of each log-type as x_{stmp} with a cost coefficient of c_{stmp} and demand constraints of the following form.

$$\sum_{smtp} a_{ismtp} x_{stmp} = b_i \quad \ldots \quad \forall_i$$

where:

x_{stmp}	is the volume of log-type p sent to mill m in period t from stand s;
a_{ismtp}	are the coefficients of each demand constraint;
b_i is	the RHS coefficient for constraint i.

The relative prices for the sub problem c_{stmp}, are then found with Equation (4.5).

$$\hat{c}_{stmp} = c_{stmp} - \sum_{i} \pi_i a_{ismtp} \tag{4.5}$$

where:

c_{stmp}	is the revenue for log-type p to mill m in period t from stand s;
$\hat{c}_{stmp}$	is the relative price used in MARVL;
π_i	are the dual variables for each demand constraint i.

The yield predictions are added to the RMP as a new proposal, and the problem re-solved.

There are two test cases reported in Cossens (1996). In the larger of the two there were 4 stands, 7 log-types and 4 mills. Minimum demand constraints were considered along with transportation cost from stand to mill. The solution process was terminated within 3 to 4 iterations when the objective was within 0.24% of the upper bound. Ninety-nine percent of solution time was spent in the yield generation subproblem. The size of the example problems was probably limited because the author needed to manually generate the input matrices to the LP solver, and transfer the data to MARVL.

4.2.2 Full OHS formulations

The full formulation of the OHS problem adds crew allocations to the bucking optimisation problem discussed previously. The addition of crew allocations adds complexity to the decisions, as only a limited number of stands can be harvested at any particular time. Harvesting crews are indivisible units, therefore an LP solution is unrealistic as the crew's location at any one time is ambiguous. When crews are considered, multi-period formulations track crew movements through time. The movement costs and penalties for crews affect profitability and production. In the discussions of the literature below, the particular details that are included in the various formulations are described. Further explanations of these details can be found in Chapter 3.

4.2.2.1 Ogweno

Ogweno (1995) presents in his PhD thesis a hierarchical planning system that includes strategic, tactical, operational, and yield prediction components.

Within the presented planning hierarchy, the OHS system (termed the operational log allocation model) contains 13 periods of a week each. The stands considered in the quarter are generated by the most recent period in the tactical plan (short term tactical plan).

Ogweno (1995) formulates the OHS model as an integer-programming problem. The problem is constructed by a PROLOG program (Clocksin & Mellish 1994) and then solved using LINDO (Schrage 1991). The integer programming formulation allocates crews to stands in each period, and movement costs are included when a crew moves from one stand to another. The harvesting capacity of the crew is included, as are the areas of the individual stands. Inventory can be carried between periods, but there is no provision for transportation costs for logs between the stands and the markets. A goal programming formulation is also presented that can help in the resolution of infeasibility.

The major difference between the Ogweno (1995) formulation and others in the literature (Weintraub et al. (1993),Murphy (1998)) is the calculation of yield predictions. It is common, in other systems in this literature survey, to generate yields based on the simulated bucking of sampled stems. In Ogweno (1995), the predicted volume of the stand is divided into *quality classes*. The quality classes are then allocated to specific log-types in the optimisation.

Quality classes represent aggregations of actual log-types, which are chosen to represent mutually exclusive subsets of log-types. Ogweno (1995) determines the volumes in these quality classes in the following manner.

- Log-types are grouped into quality classes, based on quality requirements and dimensions.
- The quality class itself is created as a log-type that is inclusive of its component log-types. Thus, a log of the quality class can be converted into a log of each of its component log-types.
- These quality classes are then ranked by their quality requirements i.e., the most stringent quality class is the highest.
- A heuristic, that sets prices, and the author's bucking simulator (Xcut) are used to determine the maximum volume of each class, effectively using a priority list bucking method.

This process and the final allocation to the log-types are shown in Figure 4.1. Note, this diagram was not included in Ogweno (1995) but was created by myself.

Actual log-type volumes are determined by constraints in the model formulation. The individual log-type volumes can be met either from volume within the log's quality class or by volume downgraded from higher quality classes, as shown in Equation (4.6)

$$q_{nit} + g_{nit} - l_{nit} - \sum_{j \in J_n} x_{ijt} = 0 \quad \forall_{int} \tag{4.6}$$

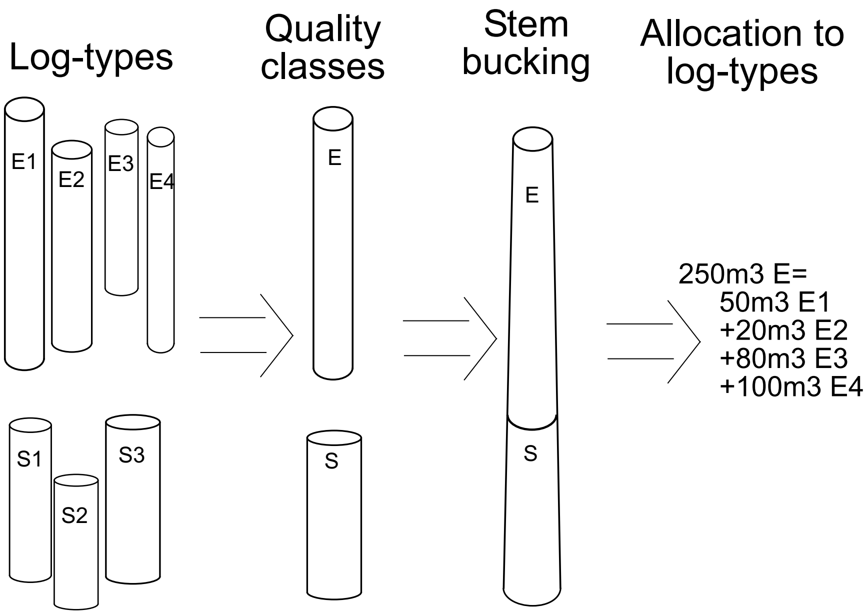


Figure 4.1: Yield calculation in Ogweno (1995)

where:

i	indexes stands;
j	indexes log-types;
n	indexes quality classes;
t	indexes periods;
J_n	is the set of logs in quality class n;
q_{nit}	is the yield of quality class n in stand i, period t;
g_{nit}	is the volume of material down graded from quality class $n-1$ to n;
l_{nit}	is the volume of material down graded to quality class $n+1$ from n;
x_{ijt}	is the volume of log-type j produced from stand i, period t.

In Ogweno (1995) the projected log-type volumes from the OHS system determines an appropriate cutting strategy to be operationally implemented. The method presented in his thesis uses stand volumes and generates a single cutting strategy that produces required volumes by altering the acceptable features and min SED requirements. This is known as *outturn control*. Interestingly, the prices are not altered.

The determination of a single cutting strategy after the OHS has been optimised is an interesting approach that is not mentioned in other literature. In other OHS systems (Epstein et al. (1999b) or Laroze (1999)), the required production from a harvest unit can be an aggregate of several different yield predictions. When operationally implementing a solution it may be difficult for harvesting crews to reconcile the varied cutting instructions.

Ogweno (1995) presents a case study using the hierarchical planning system. The case study for the OHS component is very limited, possibly because of the computational requirements to solve the integer program. The case study only includes 2 stands and 3 crews (one of which is forced to be idle). Eight log-types are considered and these form 4 quality classes. The 13 periods considered is however reasonably realistic. Because of the small size of the problem, the claim that the IP formulation is 'tight' because the IP solution is within 0.1% of the optimal LP solution is not convincing. The results of the outturn control method are also presented as successful with most volumes being within 5% of those projected, though again the small number of log-types considered does cast doubt on this conclusion.

The log-types considered in Ogweno (1995) do not include any fixed length requirements. Once fixed lengths, and more complicated log-types are added, some of the assumptions in Ogweno (1995) that underlie the quality class model of stand yields become problematic. The extra complications will make the creation of the quality classes themselves difficult.

In Ogweno (1995) there is an implicit assumption that a single downgrade path through all the quality classes exists (i.e., the volume from each quality class can be downgraded to all classes below it). For large numbers of more complex log-types this assumption is not valid.

Xcut in some cases is encased in a price generation heuristic to guarantee yields similar to a priority list. Therefore, the inclusion of a priority list bucking simulator similar to those in Laroze (1999) and Epstein et al. (1999b) would also be interesting in this system.

4.2.2.2 Murphy and Boston

Murphy (1998) describes a single period OHS model that uses Tabu search (Glover & Laguna 1993) as the solution methodology. The Tabu search heuristic allocates crews to stands with some associated yield predictions. Certain stand and crew combinations are banned, as crews can only operate in the stands where they have suitable equipment. In addition, Murphy (1998) uses *preferred stands* to indicate the present position of the crew at the beginning of the model, if a crew is not allocated to its preferred stand movement costs and penalties are applied.

Murphy (1998) details tests of the model on problems ranging from 10-60 stands, 5-10 crews (with some ability to change the maximum crews per stand), 5-8 cutting strategies and 5-

29 log-types. The solution process compared favourably with the IP solution of small problems, reaching a solution within 0.8% of the optimal IP value. This solution process was tested with variations on the Tabu value and the initial starting solution process.

Interestingly, Murphy (1998) comments about the accuracy of the data used to generate this solution and notes

> *. . . the errors associated with the data inputs are at best of the same order and at worst one or two orders of magnitude greater than the difference between the best Tabu search optimum and the theoretical optimum.*

This treatment of the errors in the input data is rare in other literature, though in my opinion particularly important for this problem.

Murphy (1998) mentions the effect of constraint 'tightness' on problem complexity. When the heuristic was tested on a real world data-set of 29 log-types with maximum and minimum demand constraints on all, it was unable to find a feasible solution. Perhaps the example problem was in fact infeasible using only the 5 cutting strategies considered, though it might have been feasible if a larger set of cutting strategies were used. It seems the type of feasibility problem illustrated in Murphy (1998), has led to the complicated column generation procedures found in Epstein et al. (1999b) and various bucking optimisation literature.

In Boston & Bettinger (1999), a forestry system is described that includes spatial inventory, logging activity (real-time harvest information system), and operational harvest scheduling. The OHS system seems very similar to the system in Murphy (1998). Additions to the formulation include the addition of multiple periods, log-stocks, and the ability to procure logs from outside sources. Boston & Bettinger (1999) uses Tabu search, but includes a genetic algorithm as a meta-heuristic to find better solutions following a procedure found in Glover et al. (1995). Boston & Bettinger (1999) reports that the procedure finds solutions between 96-99% of the optimal solution value, this is 2% improvement over a Tabu search only approach.

The formulations found in both Murphy (1998) and Boston & Bettinger (1999) are very similar to the formulation used in this thesis and discussed in Chapter 6.

4.2.2.3 Weintraub and Epstein

Work on the OHS problem has also been contributed by a group in the University of Chile. In Weintraub & Cholaky (1991) an integer-programming problem is described that decides:

- when to enter (begin harvesting) stands;
- the type of logs harvested from a stand;
- the allocation of these log-types to customers.

The integer variables control only the entry and preparation costs of the stands, though the authors mention that this requirement often leads to an integer solution for the crew locations. The problem typically covers 3 months (4 one week periods, and two month-long periods).

Harvesting capacity is constrained and total log transportation is limited by the number of trucks available. The demand for log-types is modelled in detail including transportation costs, demand levels and product specification instructions. Weintraub & Cholaky (1991) states that there are typically over 100 log-types defined. Yield predictions are produced by a product simulator that is company specific. The product simulators simulate the application of a priority list (Section 2.4.5.3) bucking pattern to the stand. This approach to yield prediction is common to all later Chilean work.

Weintraub & Cholaky (1991) solves the LP relaxation and then uses heuristics to find an integer solution. Difficulty is reported with the size of the model (386 PCs are used), and the generation of appropriate yield predictions. No case study results are published.

In Weintraub et al. (1993), an alternative method to the integer programming formulation in Weintraub & Cholaky (1991) is described. Here, expert systems (Harmon et al. 1988) are used in the solution process. Two different systems are described. One uses expert systems to determine crew and log-type allocations and to generate the necessary yield predictions. The other system uses an LP to make the allocation decisions and an expert system to generate yield predictions based on the LP dual variables.

The LP/ Expert system is similar to the iterative process described in Eng et al. (1986), where the LP solution to the crew/log-type allocation problem is used to direct the expert system to produce new yield predictions. As the yield predictions in Weintraub et al. (1993) are based on priority list bucking simulations, the dual variables from the LP cannot be used directly to generate yields as in Eng et al. (1986). The number of log-types considered in Weintraub et al. (1993) is much greater then that considered in Laroze (1999) and therefore the expert system determines both the inclusion and position of log-types on the priority list. However, Weintraub et al. (1993) does not change the log-type requirements (minimum SED, and quality required) in the priority list, so some of the flexibility of the approach in Laroze (1999) is lost.

The most impressive OHS system described in the literature is OPTICORT which is described in Epstein et al. (1999b). This system uses an LP to allocate production capacity to stands. Cutting strategies and yield predictions are generated in an iterative fashion. While the full OPTICORT model (not detailed in the paper) does allow the inclusion of different types of machinery to harvest particular groups of stands (e.g., cable crews on steep country) it does not appear to unambiguously locate the crews in each period as it remains an LP solution unlike the IP solution in Weintraub et al. (1993).

In Epstein et al. (1999b) there is a detailed discussion of the generation of yield predictions within the LP solution process. A column generation process alters priority lists based on the dual variables of the LP problem. The column generator then returns yield predictions that will improve the current solution of the LP.

Similar cost coefficients to those in Cossens (1996) are calculated from the dual variables. As the product simulators use a priority list bucking pattern, the number and order of the log-types on the list are changed (unlike models with a DP bucker that change relative values). To find the priority lists, Epstein et al. (1999b) describes a novel branch and bound procedure. This procedure iteratively determines each log-type in the priority list from the first to the last. The branch and bound procedure uses a property of priority list bucking procedures where the volume v_k of a log that is in the k^{th} position on the bucking list is independent of the logs lower down the list. This property is not true when a DP bucker is used. Again, only the inclusion and position of the logs on the priority list are changed not the log specifications, as seen in Laroze (1999).

Epstein et al. (1999b) describes and contrasts a number of different methods to navigate the branch and bound tree and shows the solution of an example problem. Two different methods of solving the column generation sub-problem are investigated. One method solves the column generation to optimality in each iteration, the other only returns the first three yield predictions that improve the solution. Epstein et al. (1999b) shows that the column generation improves the solution value by about 7% from an initial LP solution with pre-generated bucking patterns. A graph of the progress of the algorithm is shown in Figure 4.2. The number of bucking patterns considered in the example problems in this paper is very large compared to Cossens (1996) or Laroze (1999): one example has 51 bucking strategies per stand.

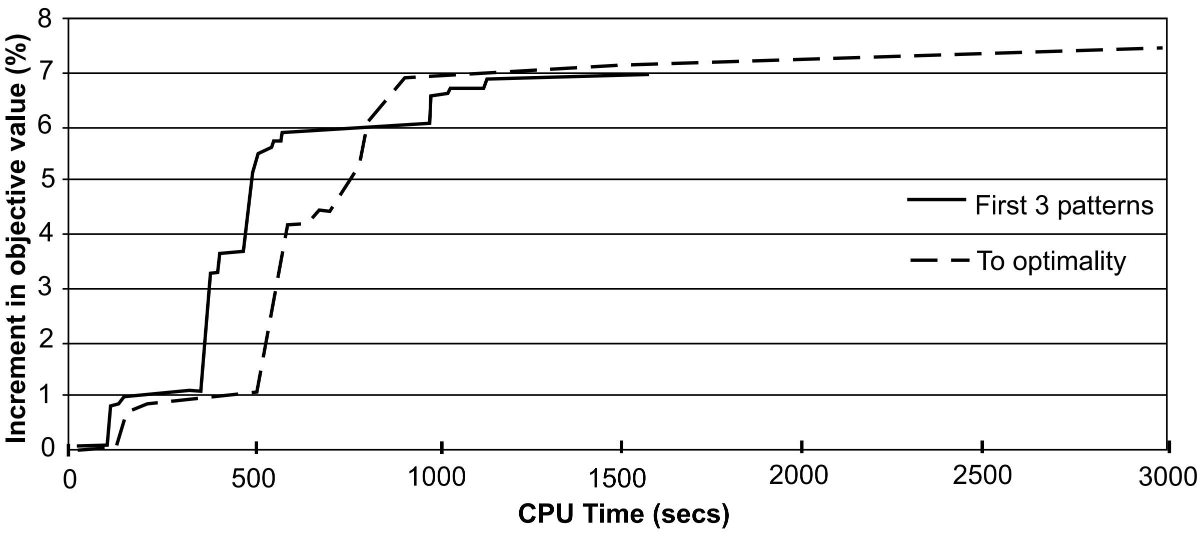


Figure 4.2: Increment in objective through time, (Epstein et al. 1999b)

The results given in Epstein et al. (1999b) are impressive because they deal with a real world sized example problem with 132 stands, 35 log-types, and initially 10 bucking patterns per stand. Unfortunately, the paper does not specify the number of periods or the size of the harvesting capacity considered. When used in Chilean forestry companies, the OPTICORT system (without the column generation) has given a 5-8% increase in net revenues compared to previous approaches. It would have been enlightening to read a comparison of a solution that uses a priority list yield predictions, with a solution that uses a DP bucker. This would quantify any loss in optimality (as an LP/DP iteration will give the optimal solution) due to the use of priority list based methods.

4.3 Conclusions

From this literature survey, conclusions can be drawn about the nature and features of a complete operational harvest scheduling model. In brief, a full model will include:

- unambiguous location of harvesting crews in each time period;
- consideration of the harvesting crews' capacities;
- multiple periods to model crew movements;
- the full forest level problem;
- iterative generation of yield predictions;

- a detailed market description, including detailed log-types, and demand restrictions;
- transportation costs from harvest unit to customer;
- some indication of the uncertainty in yield data.

Table 4.4 compares the literature using some of the above criteria. The table also shows the type of yield prediction generation used and the type of solution method.

Table 4.4: A comparison of OHS models

Model	Unambigous Location	Crew Capacity	Mult-Period	Forest Level	Iterative Yields	Yield Prediction Method	Solution Method
Mendoza & Bare (1986)				✓	✓	Stem-based Knapsack	LP
Eng et al. (1986)				✓	✓	Stem-based DP	LP
Sessions et al. (1989)					✓	Shortest Path	Binary Search
Weintraub et al. (1991)		✓	✓	✓		Priority List	Integer Program (LP+heuristics)
Laroze & Greber (1993)					✓	Priority List	Monte-Carlo Simulation
Weintraub et al. (1993)	✓	✓	✓	✓	✓	Priority List	Expert system
Ogweno (1995)	✓	✓	✓	✓		Quality Classes	IP
Cossens (1996)			✓	✓	✓	MARVL	LP
Murphy (1998)	✓	✓		✓		MARVL	Tabu Search
Laroze & Greber (1997)					✓	Priority List	Tabu Search
Laroze (1999)				✓	✓	Priority List	LP, Tabu Search
Boston & Kiser (1999)	✓	✓	✓	✓		MARVL?	Tabu Search Genetic Algorithm
Epstein et al. (1999b)		✓	✓	✓	✓	Priority List	LP Branch and Bound

Unambiguous location of crews ensures that any schedules produced can be implemented in practice. In the literature this is accomplished by using an integer programming formulation and either solving with third party MILP solver in Ogweno (1995), or by using some sort of non-optimal heuristic Weintraub & Cholaky (1991),Weintraub et al. (1993),Murphy (1998) or Boston & Bettinger (1999). None of these references discusses allowing a crew to move within a period. This movement is an interesting aspect of the OHS problem as it is possible and even desirable (see Section 3.4.2) in a solution. Standard integer programming formulations do not however allow this type of movement to be modelled.

The bucking optimisation literature focuses on the generation of a set of stand-level cutting strategies that allow the maximum value to be recovered from a forest. When market restrictions do not allow the unrestricted optimisation of individual stands, the difficulty of the problem increases. There are two methods that deal with this problem discussed in the literature: *a-priori* (before optimisation) generation of a list of suitable strategies used in Murphy (1998), Boston & Bettinger (1999), and Laroze (1999); or iterative generation within a forest level optimisation used in Eng et al. (1986), Mendoza & Bare (1986), Cossens (1996) and Epstein et al. (1999b).

The examples in the literature (Murphy (1998) and Epstein et al. (1999b)) suggest that pre-generation is ineffective for problems that contain a large number of log-types with tight market restrictions. Epstein et al. (1999b) reports a 7% increase in objective value with an iterative method. The enthusiasm for an iterative method should, however, be tempered by the realisation that these methods can easily produce solutions that can be difficult to implement operationally (Eng et al. 1986, Mendoza & Bare 1986). Pre-generation of strategies could be used to reduce computation time and to generate good initial solutions for iterative methods.

The difficulties caused by iterative methods may be overcome by a two-stage method using outturn optimisation, as recommended in Ogweno (1995). This method may provide the best mixture of the properties of iterative and a-priori methods and give solutions that can be implemented in practice.

A full market description will allow the results of the optimisation to be directly implemented in the forest. If aggregated or simplified log-types are used, or market restrictions are not modelled, the solutions will need to undergo a process of disaggregation or manual alteration in order to be operationally implemented. These processes could easily destroy any value gain from the optimisation.

Only a few references Murphy (1998),Boston & Bettinger (1999) and Epstein et al. (1999b) use more then 10 log-types in total, the rest deal with much simpler problems. In a New Zealand context a large number of log-types is required for reasons outlined in Cossens (1992).

> *New Zealand logging operations are typified by the greater numbers of log-types produced compared to other countries. This is the result of silvicultural regimes that have increased the variability of wood properties within a tree; and emphasis by marketing managers who promote niche marketing, and the manufacturing of log-types to order.*

Transportation costs can significantly affect the solution value due to the distances and the volumes involved. The OHS problem allocates supply from stands to customers and all the full OHS formulations (Murphy (1998), Boston & Bettinger (1999) and Epstein et al. (1999b)) consider transportation cost with the sole exception of Ogweno (1995). This omission may be due to the small size of the problem and the assumption that all the customers are located in the same area.

Uncertainty in yield predictions is never discussed in detail in the literature surveyed. When mentioned, it is only used as a justification for not finding truly optimal solutions as in Murphy (1998). In the description of operational planning in Gunn (1991) (see Table 4.1) the uncertainty in data is low. However, if yield predictions are solely based on inventory data there can be uncertainty around the predicted volumes, and for some particularly rare log-types, this uncertainty may be very high (this issue is discussed in Section 3.3.2). None of the literature addresses this question. In addition, none of the results discussed is compared to actual production.

The literature is divided by the use of two different algorithms for yield prediction. DP optimisation is used in Eng et al. (1986), Mendoza & Bare (1986), Sessions et al. (1989), Cossens (1996), Murphy (1998) and Boston & Bettinger (1999). Priority list simulation is found in Weintraub & Cholaky (1991), Laroze & Greber (1993), Weintraub et al. (1993), Laroze & Greber (1997), Laroze (1999) and Epstein et al. (1999b). This difference seems to be based on the country of origin of the researchers, with Chilean researchers preferring to use priority list based methods. The use of DP yield prediction methods by the New Zealand researchers in Ogweno (1995), Cossens (1996), Murphy (1998) and Boston & Bettinger (1999) would be due to the predominance of MARVL (Deadman & Goulding 1979) as the inventory tool in New Zealand industry. The use of priority list simulation by Chilean researchers may be attributed to the company specific models employed there, that are based on a priority list method. A comparison of actual production versus the prediction would be useful to justify the type of yield prediction that should be used.

Chapter 5

Optimisation Models

And here were forests ancient as the hills,
Enfolding sunny spots of greenery.
(Coleridge 1798)

In this chapter, various optimisation models that have been applied to the Operational Harvest Scheduling (OHS) problem are discussed. The differences between them are highlighted and the solution methods that are used to solve these problems are described. This chapter is intended for readers without extensive knowledge of operations research and will provide the background needed for the remainder of the thesis.

5.1 Models

This section presents some of the optimisation models used in this thesis. These include the Linear Program (LP) model, together with extensions the Mixed Integer Linear Program (MILP) and the Set Partitioning Problem (SPP). This section will describe the LP, MILP and Set Partitioning Problem (SPP) models and their solution processes.

5.1.1 Linear Programming

In the following, familiarity with Linear Program (LP) models is assumed. Background on this problem can be found in most introductory books on OR, for example Bazzaraa et al. (1990). An

LP problem is commonly solved with some form of the *Simplex algorithm* which is described in Section 5.2.1

An LP problem is represented by Equation (5.1)

$$
\begin{aligned}
\text{Minimise} \quad & z = \boldsymbol{c}^T\boldsymbol{x} \\
\text{subject to} \quad & \boldsymbol{A}\boldsymbol{x} \gtreqless \boldsymbol{b} \\
\text{and} \quad & \boldsymbol{x} \geq 0
\end{aligned}
\tag{5.1}
$$

where:

$\boldsymbol{x}$	is the vector of decision variables;
z	is the objective value of the problem;
$\boldsymbol{c}$	is the vector of cost information;
$\boldsymbol{b}$	is the vector of limits on the constraints;
$\boldsymbol{A}$	is the matrix that represents the constraints.

The important features of the LP problem are:

- all of the relationships are linear;
- there are more variables than possible equality constraints;
- the elements of $\boldsymbol{x}$ can take any real value allowed by the constraints;
- the coefficients of $\boldsymbol{A}$ are real numbers.

5.1.2 Mixed Integer Linear Programming

A Mixed Integer Linear Program (MILP) is similar to an LP but can contain integer variables. Integer variables are elements of the $\boldsymbol{x}$ vector that are restricted to integer values in a feasible solution. The inclusion of integer variables changes the nature of the feasible region in the LP problem. As the feasible region is discretized, the optimal solution may no longer be found at an extreme point. The simplex algorithm by itself is not suitable to solve the MILP problems.

The most common integer variables in a MILP problem are *binary integer variables*. These variables are restricted to a value of zero or one (hence the alternative name zero-one integer variable). These variables are frequently used to model yes-no or logical decisions. In the OHS, problem binary integer variables are used to determine whether a crew is in a harvest unit or is not.

The important features of the MILP problem are:

- all of the relationships are linear;
- there are more variables than possible equality constraints;
- some elements of $\boldsymbol{x}$ are restricted to integer values;
- the coefficients of $\boldsymbol{A}$ are real numbers.

A common method of solving MILP problems is to first solve the LP relaxation of the problem, then find integer solutions (see branch and bound method Section 5.2.3). A binary integer variable in the MILP may have fractional values in the solution to the RLP. For example, in the OHS problem, a RLP solution often indicates that a crew operates in several parts of the forest simultaneously.

5.1.3 Set Partitioning Problems

A SPP is a special type of integer problem. A set partitioning problem can be represented in the form below.

$$
\begin{aligned}
z &= \boldsymbol{c}^T\boldsymbol{x} \\
\text{subject to} \quad \boldsymbol{A}\boldsymbol{x} &= \boldsymbol{e} \\
\boldsymbol{x} &\in \{0,1\}^n \\
\boldsymbol{A} &\in \{0,1\}^{m\times n}
\end{aligned}
\tag{5.2}
$$

where:

$\boldsymbol{x}$	is the vector of integer decision variables;
z	is the objective value of the problem;
$\boldsymbol{c}$	is the vector of cost information;
$\boldsymbol{e}$	is a vector of ones;
$\boldsymbol{A}$	is the matrix that contains only zeros and ones.

The formulation is named set partitioning because the column vectors of $\boldsymbol{A}$ are associated with an element in the $\boldsymbol{x}$ variable. The $\boldsymbol{x}$ vector can therefore represent a method of partitioning the n elements of a set.

The important features of the SPP problem are:

- all of the relationships are linear;
- there are more variables then possible equality constraints;
- all of the elements of $\boldsymbol{x}$ are binary integer variables;
- the coefficients of $\boldsymbol{A}$ are restricted to zero or one.

A *set-covering problem* is formed if the equality constraints are replaced with $\geq$ constraints. Similarly, if the problem contains $\leq$ constraints it is known as a *set-packing problem*.

5.1.4 Scheduling

The application of set-partitioning problems has been especially important in the solution of scheduling problems. Scheduling problems can use a Generalised Set Partitioning Problem (GSPP) formulation which extends the SPP formulation by allowing:

- the constraints to have a mixture of equalities and inequalities;
- the RHS vector to contain non-negative integers.

A scheduling problem assigns subsets of jobs to entities that can complete them. To model a scheduling problem as a GSPP, Generalised Upper Bound (GUB) constraints are used as shown in Equation (5.3)

$$\sum_{i} x_{i,j} = 1 \ \ldots \ \ \forall j. \tag{5.3}$$

These constraints ensure that single entity (e.g., a person) can only be allocated once. Each variable in this formulation represents an assignment of an entity to a particular schedule. Thus, every column in the $\boldsymbol{A}$ matrix contributes to one and only one GUB row. Other constraints in the formulation represent the assignment of the entities to activities in the schedule.

The Crew Allocation (CA) sub-model of the Model II (see Section 6.2.2) formulation used in this thesis is an example of a GSPP. In the OHS problem, the Generalised Upper Bound (GUB) constraints force a harvesting crew to only be allocated once. Other constraints control the allocation of crews to harvest units.

A feature of a scheduling formulation is that a column represents an assignment to a complete series of activities (a *schedule*). Complicated constraints that govern the structure schedules are included implicitly by the construction of columns. These conditions could be a restriction on the total number of activities, or completion of one activity excluding performance of another. However, the number of columns in the problem can be very large, as the number of possible schedules grows in a combinatorial fashion as the number of activities increases.

In real-world problems formulated in this manner, the solution strategy cannot use an explicit $\boldsymbol{A}$ matrix that contains all possible columns. In order to solve these problems a decomposition algorithm is used called *Column generation*. Column generation (Section 5.2.2) allows the optimal solution to the problem to be found by generating columns only when they are required. A relatively small number of columns is required to guarantee optimality.

5.2 Solution methods

This section will consider three solution algorithms that are used to find the solution to the OHS problem in this thesis.

- The simplex algorithm is used to find the solution to LP problems.
- Column generation is used to generate new schedules to be considered in the OHS problem.
- Branch and bound finds integer solutions to a MILP given the solution to its RLP.

5.2.1 Simplex

The *Simplex algorithm* was first described by Dantzig in 1947 (Dantzig 1963). The simplex algorithm solves an LP problem by searching the extreme points of the feasible region (or simplex). An outline of the general simplex algorithm is given in Algorithm 5.1. Briefly, the simplex algorithm begins with an initial *basic feasible solution*. It then determines which variable not in the *basis* (the *entering variable*) is pivoted into the solution to improve the objective value. A variable must leave the basis in the pivot, this variable is determined from the *leaving variable* criteria. The simplex algorithm continues until one of following three results.

1. No initial feasible solution is found therefore the problem is infeasible.

2. No leaving variable can be found therefore the problem is unbounded.

3. No entering variable is found therefore the current solution is optimal.

Algorithm 5.1 Simplex algorithm

```
Find the A matrix and b vector from the problem formulation
Partition A = (B|N) and x = (x_B, x_N)
Require: B and x_B represent an initial feasible solution
while Problem is neither optimal or unbounded do
  Find an entering variable from x_N
  if no ev is found then
    Problem is optimal
  else
    Find the leaving variable from x_B
    if no lv is found then
      Problem is unbounded
    else
      Replace lv with ev in B
    end if
  end if
end while
```

5.2.1.1 Initial basis

To begin the simplex algorithm an initial basic feasible solution is required. Constructing a feasible basis can be difficult when the problem has many constraints. One effective method to find an initial feasible basis is a phase-one phase-two method. In this method *artificial variables* are introduced into the problem. These artificial variables are columns of the identity matrix and allow the initial basis to be formed as the identity matrix.

To remove the artificial variables, the objective is changed in phase one. The artificial variables have a cost of one, and the original variables cost nothing. The simplex algorithm is then used to remove the artificial variables. When the objective value in phase one becomes zero all artificial variables have left the basis, or are at zero value. At this point the solution method switches to phase two and the costs are returned to their normal values. The simplex algorithm proceeds from this point as a feasible basis has been found.

5.2.1.2 Choice of entering variable

The entering variable in the simplex algorithm is chosen to improve the objective value of the problem. A measure of the improvement caused by the inclusion of a variable in the basis is the *reduced cost* (RC) of the variable.

To find the reduced cost we define the vector $\boldsymbol{c_B}^T\boldsymbol{B}^{-1}$ as the $\boldsymbol{\pi}$ (pi) vector or the dual variables of the LP

$$\boldsymbol{\pi}^T = \boldsymbol{c_B}^T\boldsymbol{B}^{-1} \tag{5.4}$$

where:

$\boldsymbol{c}_B$	is the vector of costs of variables in the basis;
$\boldsymbol{B}$	is the basis matrix.

The reduced cost of a variable x_s is

$$RC = c_s - \boldsymbol{\pi}^T\boldsymbol{a}_s \tag{5.5}$$

where:

$\boldsymbol{a}_s$	is the vector in $\boldsymbol{A}$ associated with x_s;
c_s	is the cost of variable x_s.

This calculation of the reduced cost and the dual variables is important in the column generation algorithm.

The standard simplex algorithm finds the entering variable x_s from Equation (5.6)

$$s \in \arg\min_{i\in\boldsymbol{N}} (c_i - \boldsymbol{\pi}^T\boldsymbol{a}_i : c_i - \boldsymbol{\pi}^T\boldsymbol{a}_i < 0). \tag{5.6}$$

5.2.1.3 Calculation of leaving variable

Once an entering variable is selected by the reduced cost criterion, the variable that will leave the basis (leaving variable) in the pivot is found. The variable that leaves the basis is the first variable to become negative as the entering variable increases. The change in the basic variables for a unit increase of the entering variable is found from the vector $\boldsymbol{B}^{-1}\boldsymbol{a}_s$.

The leaving variable is found from variables in $\boldsymbol{x}_B$ that correspond to negative elements of

$\boldsymbol{B}^{-1}\boldsymbol{a}_s$. The leaving variable has the minimum ratio calculated in the leaving variable criterion

$$p \in \arg\min_{i \in \boldsymbol{B}} \left(\frac{\boldsymbol{e}_i{}^T \boldsymbol{B}^{-1}\boldsymbol{b}}{\boldsymbol{e}_i{}^T \boldsymbol{B}^{-1}\boldsymbol{a}_s} : \boldsymbol{e}_i{}^T \boldsymbol{B}^{-1}\boldsymbol{a}_s > 0 \right) \tag{5.7}$$

5.2.1.4 Stopping conditions

The simplex method continues iterating until one of three conditions occurs.

- No new entering variables are found.
- No leaving variables are found.
- The basis is degenerate and the simplex algorithm is stalling.

If no new entering variable can be found, the current solution is declared optimal. The simplex method stops and reports the optimal solution.

If no leaving variable is found the problem is then unbounded. In an unbounded problem a single variable can increase infinitely while the objective value improves, therefore, the feasible region is unbounded. In real world problems, an unbounded solution usually indicates that the model formulation is incorrect.

If the problem is degenerate, the simplex method can cycle through many iterations without improving the objective value. Unfortunately, GSPP problems are susceptible to degeneracy because of the large number of variables at zero in a basic feasible solution. Therefore, techniques to resolve degeneracy are implemented in software that solves GSPP problems. An example of an algorithm that deals with stalling is Wolfe's method (Wolfe 1963, Ryan & Osborne 1988).

5.2.2 Column generation

The formulation of a GSPP necessitates a large number of variables in an explicit formulation. For example, the Model II formulation (Section 6.2.2) of the OHS problem could have as many as 1.68×10^{22} possible crew schedule variables in a realistic problem[1]. Clearly, this is an excessive number of variables to consider in the simplex algorithm. However, a column generation algorithm can be used to generate a sub-set of variables that are sufficient for the optimal solution of the GSPP problem.

[1]These figures are based on 10 crews, 30 harvest units with 15 cutting strategies each and 8 periods.

Column generation is an example of a decomposition algorithm. It was first used to solve the cutting stock problem in Gilmore & Gomery (1965). Figure 5.1 shows the column generation process.

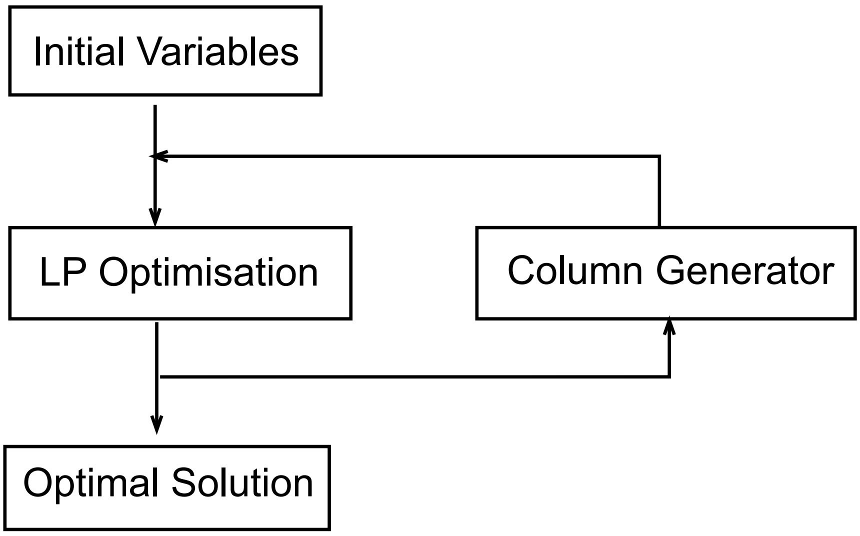


Figure 5.1: The column generation process

Specifically the GSPP problem can be broken in two, the Restricted Master Problem (RMP), and the Column Generation Algorithm (CGA). The RMP only considers a small subset of the possible variables. These initial variables are chosen *a priori* at the beginning of the algorithm. The selection and construction of the initial variables could be very simple, or may take into account the complexities of the specific problem and attempt to provide a 'good' initial solution to the problem.

Given the set of initial variables $\boldsymbol{P}$ the RMP is

$$\begin{aligned} \text{Minimise} \quad & z = \boldsymbol{c}^T \boldsymbol{x}_P \\ \text{Such that} \quad & \boldsymbol{P}\boldsymbol{x}_P = \boldsymbol{b} \\ & \boldsymbol{x}_P \geq 0. \end{aligned} \tag{5.8}$$

The optimal solution of this RMP is not the optimal solution to entire problem as it does not consider all of the possible variables. However, the reduced cost of an entering variable is defined as $c_a - \pi^T x_a$ from Equation (5.5).

The sub problem in the column generation is defined as

$$\begin{aligned} \text{Minimise} \quad & rc = c_a - \boldsymbol{\pi}^T \boldsymbol{a} \\ \text{Such that} \quad & \boldsymbol{a} \quad \text{is a feasible variable to be introduced} \\ & \qquad \text{into the master problem, with cost } c_a. \end{aligned} \tag{5.9}$$

The solution to Problem (5.9) is used to find an entering variable to the restricted master problem. If the optimal value of rc (rc^*) in Problem (5.9) is a negative value ($rc^* < 0$) then $\boldsymbol{a}$ enters the basis of the RMP and the simplex algorithm continues. However, if rc^* is non-negative ($rc^* \geq 0$) then the variable $\boldsymbol{a}$ is not an entering variable to the RMP. The column generation sub-problem should be able to consider all legal variables in the master problem, if no feasible columns are found with a negative reduced cost, the master problem can then be declared optimal.

The column generation algorithm can be more efficient if it returns more than one variable in each iteration. If it does, the number of times the column generation algorithm is called should be reduced. In addition, it can be more efficient to return a variable with negative reduced cost but not necessarily find the optimal solution to Problem (5.9). Any technique can be used in the column generation as long as a negative reduced column is returned, if one exists.

Problem (5.9) is itself an optimisation problem. The sub problem can be solved by a variety of methods depending on the type of problem. Linear Programming, dynamic programming, assignment and travelling salesman type formulations are possible.

5.2.2.1 Dynamic programming

The columns of the Model II formulation of the OHS problem represent the sequence of harvest units visited and the cutting instructions for each period. Variable construction is discussed at length in Chapter 6. The column generation sub-problem (Problem (5.9)) is formulated as a shortest path problem and a dynamic programming recursion is used to find its solution.

Dynamic programming is particularly suited to finding solutions to problems with some linear ranking, as this can preclude cycles. In the OHS problem, that ranking is given by the time dimension of the problem. A cycle is not possible as a crew cannot return to a previous time period. Other problems where dynamic programming has proven useful involve shortest path problems (i.e., the Bellman-Ford algorithm), an example is stem bucking problems (Section 2.5.1.4).

Dynamic programming can be used to solve linear and non-linear problems and is commonly applied to problems with integer features. A stage in a DP is defined as the progress through the ranking, e.g., the stages in the OHS problem are the periods. A state is defined as a resource or quantity that is affected by the decisions made in a DP model. In addition to stages and states, the DP formulation requires an objective function so the quality of solutions can be compared.

The requirement for a problem that allows DP as a solution approach, is that the problem obeys the DP *principle of optimality*

DP: 1 (Principle of optimality) *The decision made at each state and stage is not affected by decisions in previous stages*

Consider a DP recursion that solves a shortest path problem. If the best paths to each state in stage t-1 are found, a DP recursion can use this principle to find the best path to state **X** in stage t. The cost of moving from each state in t-1 to state **X** is evaluated. The minimum cost path to state **X** and stage t is then recorded. As the past decisions do not affect future decisions, only the state, stage and objective value of previous stage's paths are considered.

For example, a problem has four stages and three states (**A, B, C**) in each stage. The total number of possible candidate solutions is $3^4 = 81$. A dynamic program will only evaluate 9 candidates (3 states in the previous stage $\times$ 3 states in the current) at three stages. The total number of evaluations is $3^2 \times 3 = 27$. There are also additional savings in time as the evaluation in the early stages will involve less computation.

A number of techniques can improve performance of a typical DP recursion. The cost evaluation is performed often in the recursion so improvements in this part of the algorithm significantly increase performance. An effective way to improve performance is to pre-calculate as many of the cost factors as possible before the DP recursion. Pre-calculation can significantly improve performance when the same calculation is required many times in the recursion.

5.2.3 Branch and Bound

To find a solution to an integer programming problem, a Branch and Bound (B&B) algorithm is applied to the linear programming relaxation of the MILP problem. The Relaxed Linear Pro-

Algorithm 5.2 Dynamic programing recursion

$f_{i,j}$ the value of state j at stage i
$p_{i,j}$ the predecessor of state j at stage i
$c_i(k, j)$ cost of moving from state k to state j at stage i
Require: $f_{0,j} = c_0(j)$
Require: $p_{0,j} = \theta$
  for $i = 0 \rightarrow i_{max} - 1$ **do**
    for $j = 0 \rightarrow j_{max}$ **do**
      $f_{i+1,j} = \max\limits_{k} (f = f_{i,k} + c_i(k, j))$
      $p_{i+1,j} \in \arg\max\limits_{k} (f = f_{i,k} + c_i(k, j))$
    end for
  end for

gram (RLP) is formed by removing integrality requirements from a MILP formulation. Branch and Bound is a structured way of exploring the feasible region of an integer program. At each stage of the B&B tree, a decision is applied that divides the feasible region (normally into 2 parts), based on some property of the integer solution that is not present in the linear relaxation.

In a *variable branch*, a single variable from the problem is taken and (if it is a zero-one variable) forced to zero on one side of the branch and one on the other. This approach exploits the binary integer property of the variable.

In a *constraint branch*, a decision is made that affects a large set of binary variables. A constraint branch is either a one-branch, where the decision is enforced and all variables that do not comply are removed; or a zero-branch which precludes the branch decision.

When a feasible integer solution is found for some node in the B&B tree, other nodes in the tree may be bounded or removed from further consideration. When a branch is applied to a node the objective value of any child nodes cannot be better then the parent node's objective. Therefore, if the objective of any node is worse than the best integer solution found the node will not be considered further.

5.2.3.1 Node choice within Branch and Bound

The *node choice* while traversing a branch and bound tree affects the efficiency and effectiveness of the solution strategy. Many alternative strategies can be applied, two main strategies are *depth-first branching* and *breadth-first branching*. Depth first branching evaluates a node in the tree and in the next stage evaluates one of its children. This strategy tends to find an integer solution quickly or forces the nodes to be infeasible. If an integer solution is quickly found, the bound applied helps reduce the effort in traversing the rest of the tree.

Breadth first branching chooses the node with the lowest objective value at each stage and will tend to explore a B&B tree one level at a time. The first integer solution found by this method may have a good objective value, near the optimal integer value. However, breadth first branching may take a long time to find an integer solution.

5.2.3.2 Bound tolerance and integer allocation

Another technique that finds solutions quickly within branch and bound, is the use of a *bound tolerance*. With this technique, a tolerance (for example $\leq 5\%$) is applied to the objective values used to bound nodes. A node is bounded if its objective is worse than the best integer solution plus the bound tolerance. If an integer solution is found with an objective within the bound tolerance of the RLP objective, the algorithm reports this solution is the optimal integer solution.

The objective values used to bound the nodes are not necessarily produced by branching. Any objective from a feasible integer solution can be used to produce a bound on the active nodes in the tree. Thus, any heuristic solution can be used to bound the tree search. A simple heuristic that finds a good quality integer solution from an example nodal solution aids the branch and bound process enormously. This technique is known as *integer allocation*.

5.2.3.3 Constraint branching

Constraint branching is a method of branching that compensates for some of the deficiencies of variable branches applied to set partitioning problems. In Ryan (1992), the concept is explained with reference to problems in airline crew scheduling formulated as SPP. Constraint branching in this application is designed to reduce the sequence depth of the problem (Ryan & Falkner 1988) and thus give the problem desirable properties (uni-modular, perfect, balanced) that produce naturally integer solutions. In this case, a constraint branch is defined in Algorithm 5.3.

A one-branch forces a single variable to satisfy both constraints while a zero-branch forces the constraints to be satisfied by two separate variables. Note, neither branch affects variables that are inactive in both constraints.

Algorithm 5.3 Constraint branching algorithm

$a_{n,m}$ is an element of the $\boldsymbol{A}$ matrix
Find two constraints on rows i, j in the SPP
$i \in J \iff a_{i,j} = a_{i,k} = 1$ {i is active in both constraints}
$i \in \bar{J} \iff a_{i,j} \neq a_{i,k}$ {i is only active in one constraint}
$\sum_{i \in J} x_i = 1$ {One-branch}
$\sum_{i \in \bar{J}} x_i = 0$ {Zero-branch}

In work published after Ryan & Falkner (1988) constraint branches have been generalised. In general, a constraint branch divides the problem variables into two sets based on the component decisions within the variables. Literature has used various branching decisions that are based on the specific problem formulations. In McNaughton (1998)[2], the branches are defined as decisions to harvest sections of forests accessed by specific roads.

The generalised properties of a constraint branch are described below.

- Constraint branching if applied repeatedly will eventually force an integer solution or create an infeasible problem.
- All feasible integer solutions can be found by the application of a series of constraint branches.
- A constraint branch forces groups of variables out of the solution on either side of the branch, making it more 'balanced' than a variable branch.
- A constraint branch forces integer properties onto the constraint matrix.

The criteria used to constraint branch in this problem are discussed in Section 8.3.2.

[2] A Phd. thesis that describes the solution of forestry tactical planning problems.

Chapter 6

Problem Formulation

But oh! that deep romantic chasm which slanted
Down the green hill athwart a cedarn cover!
(Coleridge 1798)

In this chapter the mathematical formulation of the Operational Harvest Scheduling (OHS) model is discussed. This model is derived from the OHS problem described in Chapter 3. The assumptions and simplifications in this model are detailed. Several different methods of formulating the decision variables, including the formulation most often used in literature, are described and contrasted with the formulation used in this thesis.

6.1 Model definitions

In this chapter, some terms are used to describe the following concepts.

- *Crew allocation*: The decision to place a particular crew in a harvest unit in a period.
- *Crew schedule*: A sequence of crew allocations that controls a crew's movements from the beginning to the end of the planning horizon.
- *Schedule*: A sequence of harvest units and periods, that is defined independently of any crew.

The OHS problem is described in Section 3.1.1. A general statement of the OHS optimisation problem is:

Maximise	Profits (revenue-costs)
While	Meeting volume constraints
By allocating	Crews to harvest units
	with cutting strategies;
	and allocating
	production to customers;
	throughout the time horizon.

6.1.1 Division into sub-models

In this thesis, a distinction is made between the *Crew Allocation (*CA*) sub-model* and the *Production/ Transportation (*PT*) sub-model*. The CA sub-model governs the crew allocations and consists of the following constraints:

- crew assignment (Section 6.4.1);
- harvest unit capacity (Section 6.4.2.3);
- harvest unit area (Section 6.4.2.4).

The CA sub-model also contains the variables that control crew allocations. The constraints of the CA sub-model form a GSPP submatrix (see Section 5.1.3) with the addition of the harvest unit area constraints. A crew cannot exist in two separate locations simultaneously. Therefore, decisions on crew allocations are usually modelled as binary integer decision variables.

The allocation of production and other decisions discussed in Section 6.5 are modelled in the PT sub-model. The PT sub-model comprises the other constraints and variables given in the formulation (see Section 6.5). Within this sub-model is a transportation problem and the decisions are modelled as continuous (i.e., not binary) decision variables.

The linkages between the two sub-models are the volume allocation constraints (see Section 6.5.1). The combination of binary integer and continuous variables in the two sub-models, make the mathematical model of the combined problem a type of Mixed Integer Linear Program (MILP) (see Section 5.1.2).

The following section describes in detail the overall structure of the model particularly focusing on the form of the CA sub-model.

6.2 Possible structures

The OHS problem may be modelled in at least five ways (see Sections 6.2.1–6.2.3.3). Two of these models are discussed in detail, while the other three are included for completeness. The crucial difference between the models is the form of the CA sub-model. A single binary integer variable can represent a single crew allocation (in a Model I formulation), or a composite of several crew allocations (in Models II-V).

In the following discussion, only the CA sub-model will be detailed. However, the form of the PT sub-model will be subtly affected by the choice of model formulation. This is discussed in detail in Section 6.5.

To simplify the model comparison, the choice of cutting strategy is not discussed. In each of the models, the cutting strategy decision will be combined together with the harvest unit choice. This assumption is valid however, as the yield predictions (see Section 2.4.5) are dependant on a combination of cutting strategy and harvest unit. Therefore, these two decisions should be made together. Five possible models are listed below.

- *Model I*: Each crew, harvest unit (/cutting strategy) and period combination is modelled as a separate binary integer variable. There are no composite decisions. In the integer solution, one variable is positive for each crew in each period.
- *Model II*: The binary integer variables allocate a crew to a sequence of harvest units. The sequence of harvest units throughout the time horizon becomes a composite decision. In the integer solution, a single variable is positive for each crew.
- *Alternative Formulations*
  - *Model III*: The binary integer variables represent all crew harvest unit assignments in each period. The crew harvest unit allocation is a composite decision. In the integer solution, there is a single positive variable for each period.
  - *Model IV*: The binary integer variables represent a harvest unit allocated a sequence of crew visits. The crew visits over time are collapsed into a single decision. The integer solution will have a positive variable for each harvest unit.
  - *Model V*: All the crew harvest unit and period decisions are modelled as a single binary integer variable. Different variables represent a different set of decisions. Thus, all the integer decisions become a single composite decision. In the integer solution, only a single binary variable will be positive. However, the decisions in the PT sub-model will remain separate.

These formulations are described in detail below. Reference is also made to the crew assignment constraint (Section 6.4.1) and the harvest unit capacity constraint (Section 6.4.2.1). The motivation for the constraints is discussed in detail in the relevant sections but they are presented here so that the differences between the models can be clearly explained.

In the following discussion, the number of variables in each formulation is compared. Reference is made to an explicit formulation where each possible variable in a formulation is explicitly represented. Examples of this representation are data formats such as a *.MPS* file. In this thesis however, the chosen formulation is never explicitly represented since a column generation algorithm (Section 5.2.2) is used. Only a subset of possible variables is ever represented. Another comparison made between formulations is the number of binary integer variables that are positive in an integer solution. This number is largely controlled by the nature of the Generalised Upper Bound (GUB) constraints in each formulation.

In the following discussion, only the binary integer variables in the formulations are considered. The continuous variables form part of the PT problem and are therefore disregarded within this section.

6.2.1 Model I: Crew, harvest unit, period

A *Model I* formulation has a single binary variable for each crew allocation.

A decision variable x_{cht} is associated with crew c allocated to harvest unit h (and therefore a cutting strategy) in period t (periods are typically one week long),
where:

- c indexes the crews;
- h indexes the harvest units;
- t indexes periods.

Thus $x_{1,A,1}$ represents the allocation of crew 1 to harvest unit **A** in period one.
The crew constraint is modelled as

$$\sum_h x_{cht} = 1 \ldots \forall c, t$$

Note, in this thesis $\forall c$ indicates that the constraint is repeated once for each crew in the problem and is equivalent to $\forall\{c \in C\}$

where:

C is the set of crews.

The harvest unit constraint is

$$\sum_{c} x_{cht} \leq G_h \ldots \forall h, t$$

where:

G_h is the maximum number of crews allowed in harvest unit h.

Model I formulations have previously been used to formulate OHS problems in the forestry literature. It is the model used in Ogweno (1995), CONDOR in Murphy (1998), Boston & Bettinger (1999) as well as the OPTICORT system in Epstein et al. (1999b) (see Section 4.2.2 for details). An advantage of this formulation is that the number of integer variables created in an explicit formulation is limited to $|C||H||S||T|$

where:

$|C|$ is the total number of crews in the problem;

$|H|$ is the total number of harvest units in the problem;

$|S|$ is the total number of cutting strategies in the problem;

$|T|$ is the total number of periods in the problem.

This modelling form contains the least number of variables in the explicit formulation but has the greatest number of positive variables in the integer solution. A difficulty for this formulation is the lack of an explicit link between any of the decisions. The crew allocation in one period is difficult to link to the crew allocation in the next period.

An example problem can be used to compare the different model formulations. This problem has 10 crews, 30 harvest units with 15 cutting strategies each and 8 periods. A Model I formulation will have a total of $36,000$ possible variables for this example problem.

This figure is well within the limits of current computing capacity for LP problems. Therefore, it is not necessary to use column generation in this formulation. Without column generation, standard mathematical programming languages and solvers can be used, for instance, AMPL(Fourer et al. 1993), GAMS(Brooke et al. 1992) or LINGO(Scharge 1993).

6.2.2 Model II: Crew to harvest unit sequence

In a *Model II* formulation, each variable represents a *crew schedule*. A crew schedule determines a crew's location and what logs it harvests in every period of the problem. The combination of activities for each crew in the time horizon is similar to set partitioning formulations that have been used to solve scheduling problems as outlined in Section 5.1.3.

A decision variablex_{ci}, is associated with crew c allocated to schedule i where:

i indexes possible schedules.

For instance $x_{1,123}$, is crew one following schedule 123. Schedule 123 is an example of one of the possible crew schedules. Crew schedule $x_{1,123}$ could represent the decisions shown in Table 6.1.

Table 6.1: Example of a variable for a Model II formulation

Crew 1			
	Period	1	A
	Period	2	A
	Period	3	A
	Period	4	B

This represents crew 1 visiting harvest unit **A** in periods one to three and then moving to harvest unit **B** in period four. Note, again the allocations of cutting strategy have been omitted for simplicity.

The crew constraint is modelled as

$$\sum_i x_{ci} = 1 \ldots \forall c$$

where:

c indexes the crews;

i indexes possible schedules.

The harvest unit constraint is

$$\sum_{c,i} H^i_{ht} x_{ci} \leq G_h \ldots \forall h, t$$

where:

H^i_{ht} $=1$ when schedule i contains harvest unit h in period t and zero otherwise.

In this model, a single variable represents a crew's activities throughout the planning horizon. Therefore, movement costs can be easily included in the variable's cost. The effect of movement on productivity can also be calculated. Crew schedules can be removed from the problem if they represent solutions that could not exist in an integer solution (discussed in detail in Section 7.2). The RLP (see Section 5.2.3) solution will therefore be closer to the integer solution than is possible in a Model I formulation.

The number of variables in an explicit formulation of this problem is very large because of the combination of the harvest units, cutting strategies and periods in each crew schedule variable. The total number of possible variables is $|C|(|H||S|)^{|T|}$. In the example problem mentioned previously with 10 crews, 30 harvest units, 15 cutting strategies and 8 periods there will be 1.68×10^{22} variables .

Because of the large number of variables, an explicit formulation is not possible for anything but very small problems. To solve reasonably sized problems with this formulation, column generation techniques (Section 5.2.2) are used.

The column generation subproblem can be posed as a shortest path problem. Each node represents a harvest unit allocation in a period, and arcs connect the nodes to those in the previous period. The paths described represent the sequence of harvest units visited by a crew. This network can consider the penalties and costs of moving the crew between the harvest units, as well as any other factors that rely on the history of the crew's previous position.

The Model I and II formulations form the basis of further discussion in this thesis. A detailed comparison of these two formulations can be found in Section 6.2.4. The additional model formulations Models III through V are presented here for completeness and theoretical consideration only, and are not discussed further in this thesis.

6.2.3 Alternative formulations

6.2.3.1 Model III: Period to crew assignment

In the *Model III* formulation crew harvest unit allocations are grouped together and assigned to a period.

So a variable will be x_{tk}, which represents the crew to harvest unit allocation k in period t where:

k indexes possible crew harvest unit allocations.

For example, an instance of this variable $x_{1,123}$ is the decision shown in Table 6.2.

Table 6.2: Example of a variable for a Model III formulation

Period 1

Crew	1	A
Crew	2	B
Crew	3	D
Crew	4	G
Crew	5	C

This represents in period one: crew one will operate in harvest unit **A**; crew two in **B**; crew three in **D**; crew four in **G**; and crew five in **C**.

A period constraint ensures there is an allocation for each period in the problem

$$\sum_k x_{tk} = 1 \ldots \forall t.$$

In this formulation, the crew constraint is implicit in the construction of the columns, as each legal variable will allocate a crew to a harvest unit. The above period constraint will therefore guarantee a valid crew assignment.

The harvest unit constraint is also contained in the column construction as the number of crews allocated to single harvest unit will not exceed G_h in each valid crew harvest unit allocation k.

The movement costs and penalties will be difficult to implement.

The total number of integer variables is $|T|(|H||S|)^{|C|}$. In total there will be 2.72×10^{27} variables in the example problem. However, this total number will be reduced by the constraints on valid variables mentioned above.

As in Model II, an explicit representation of this formulation will be impractical, so column generation must be used to solve the problem. In contrast to Model II, each column represents the assignment of crews to the harvest units in a single period. The column generation subproblem can be posed as an assignment problem (not a shortest path as in Model II). This formulation will be useful when considering factors that relate to the allocation of crews in a single period e.g., if crews need to work in harvest units adjacent to other crews, or if crews cannot work near each other.

6.2.3.2 Model IV: Harvest unit to crew sequence

The *Model IV* formulation is similar to Models II and III. For each harvest unit a schedule of the crews that occupy it in each period is a binary integer decision variable.

The variable is $x_{h,q}$
where:

q indexes the crew sequences available for a harvest unit.
For example, variable $x_{B,123}$ represents the decision in Table 6.3.

Table 6.3: Example of a variable for a Model IV formulation
Harvest unit B

Period	1	Crew	2
Period	2	Crew	2
Period	3	Crew	2
Period	4	Crew	1

Table 6.3 indicates harvest unit **B** is harvested by crew 2 for the first three periods, then crew 1 for the fourth period.

The harvest unit constraint is

$$\sum_q x_{hq} \leq G_h \ldots \forall h.$$

The crew constraint becomes

$$\sum_{h,q} C^q_{ct} x_{hq} = 1 \ldots \forall c$$

where:

C^q_{ct} $= 1$ if schedule q contains crew c in period t.

Note, in this model the cutting strategy decision is grouped with the crew decision and not included with the harvest unit decision as in the other models.

As in Model II, the column generation subproblem will again be posed as a shortest path problem. A path will represent the sequence of crews that will harvest a particular harvest unit.

The number of integer variables required will be $|H|(|C||S|)^{|T|}$. Therefore, there are 7.69$\times 10^{18}$ variables in a realistic problem.

Many of the harvest unit variables will have no crews allocated to them for the entire time horizon in the shorter-term planning problem. While not well suited to the stated OHS problem, if several crews were required to visit a harvest unit in sequence this formulation would work well.

6.2.3.3 Model V: Combined decisions

In *Model V*, the decisions are removed entirely from the integer-programming problem, and placed in a single composite variable. In fact, the master problem serves only as a method of costing and choosing between the full schedules and satisfying the PT constraints.

A binary decision variable is x_r

where:

r indexes possible complete solutions to the CA problem.

An instance of the variable x_{123} could represent the solution (note the dots indicate decisions omitted for brevity) in Table 6.4. This variable details the location of all the crews for each period in the planning horizon.

The only CA constraint needed in this formulation is

$$\sum_r x_r = 1$$

Table 6.4: Example of a variable for a Model V formulation

Crew	1	Period	1	A
Crew	1	Period	2	A
Crew	1	Period	3	A
Crew	1	Period	4	B
Crew	2	Period	1	B
…	…	…	…	…
Crew	2	Period	4	D
…	…	…	…	…
…	…	…	…	…
Crew	5	Period	4	F

All the other integer constraints can be included implicitly in the generation of valid columns.

This formulation effectively solves the CA problem within the column generation subproblem, with the master problem providing dual variables and refining the previous solutions from the column generation.

The number of integer variables required is $(|H||S|)^{|T|^{|C|}}$. Therefore, there are 1.81×10^{212} variables in a realistic problem, a truly staggering number.

This formulation contains the largest number of binary integer variables and has the least number of positive variables in the integer solution (one).

A significant disadvantage of this problem formulation is that an additional optimisation heuristic will have to find the suitable columns. As a single column contains all of the CA decisions only the PT decisions are made in the LP .

6.2.4 Model comparison

Of these models, Models I and II are the most important. Model I describes the formulation that is found in the literature, while Model II describes the novel method used in this thesis.

Simple one-period problems, stated in both Model I and II formulations are identical. However, in multi-period formulations Model II formulations are preferable as the movement costs are included in the variable construction. In a Model I formulation movement costs can be considered but need to be implemented with extra constraints and variables. In addition, the techniques have been developed for the Model II formulation in my research allows crews to change harvest units mid-period (Section 7.1) and allows a tighter statement of the RLP than is possible in Model I (Section 7.2).

An advantage of Model II solutions is evident in schedules that contain longer time horizons where the crews move more often. In these problems, all the crews will need to move at least once and perhaps several times. A Model I formulation becomes increasingly difficult to solve in these situations while the complexity of a Model II formulation increases at a lesser rate.

Interestingly, constraint branching (Section 5.2.3.3) in the Model II formulation (see Section 8.3) makes identical operational decisions[1] to variable branches in a Model I formulation. A constraint branch in Model II chooses a set of crew schedule variables that share a common crew allocation and bans crew schedules without this allocation. A crew allocation is represented by a single variable in Model I, therefore a variable branch keeps or removes this crew allocation. The use of constraint branching allows both the advantages of a Model II problem formulation and the limited branch and bound tree generated in a Model I formulation.

A Model III formulation will be useful in situations where the relative locations of crews in each period are important. For example, if crews need to remain near each other, or they need to remain separated.

A Model IV formulation will be advantageous when there are relatively few harvest units that must be visited numerous times by different crews in some specific order. Perhaps silvicultural operations (for instance, planting and pruning) could be scheduled in this manner.

A Model V formulation could be appropriate if there are significant restrictions on possible schedules that limit the size of the column generation problem. However, for this formulation to be effective the column generation will have to be efficient. A useful application of this formulation will be to embed a non-optimal heuristic (perhaps the tabu search in Murphy (1998)) to generate candidate columns. The LP formulation will then determine volume allocations.

The next section discusses the construction of a Model II formulation.

6.3 Construction of constraint matrix

A simplified overview of the matrix is shown in Figure 6.1. An additional large format copy of this matrix is found in Appendix A. The matrix shown represents a problem that includes

[1]However, these approaches are significantly different in terms of solution strategy implementation.

three crews, two log-types, two customers and two periods. Note, that only four example crew schedules are shown, as there are too many possible schedules to list them all. The illustration also only includes the minimum market constraints. For instance, the minimum demand is shown and the maximum demand is omitted.

The various constraints and variables in this matrix will be discussed in detail in the following sections. Figure 6.1 can be used to see the overall structure of the matrix while the detail is discussed.

In Figure 6.1, the CA sub-model can clearly be seen as it consists of the first three sets of constraints. The PT sub-model completes the matrix. The linkages between the two models are the contributions to the volume allocation constraints (see Section 6.5.1).

The CA sub-model essentially defines an OHS problem. The PT constraints however, can often vary depending on the practices of the particular forestry company. In this thesis, the properties of the CA sub-model are used to define the constraint branching and column generation. The PT sub-model provides input to these algorithms (e.g., the input prices to the column generation), and does not affect the solution process. This separation allows the PT sub-model to vary in different forestry companies, while the solution strategy outlined in this thesis remains valid.

A crew schedule is a combination of a crew c with a schedule. The schedule i is a sequence of harvest units and cutting strategies. To extract information from a schedule two coefficients are defined.

H^i_{ht} $= 1$ when schedule i contains harvest unit h in period t and zero otherwise;

S^i_{st} $= 1$ when schedule i includes strategy s in period t and zero otherwise.

Note, that these are not decision variables themselves but represent component decisions within the composite decision variable x_{ci} in a Model II formulation.

In addition, these coefficients may be referenced as functions.

$H(i,t)$ is the harvest unit harvested in period t by schedule i;

$S(i,t)$ is the strategy used in period t by schedule i.

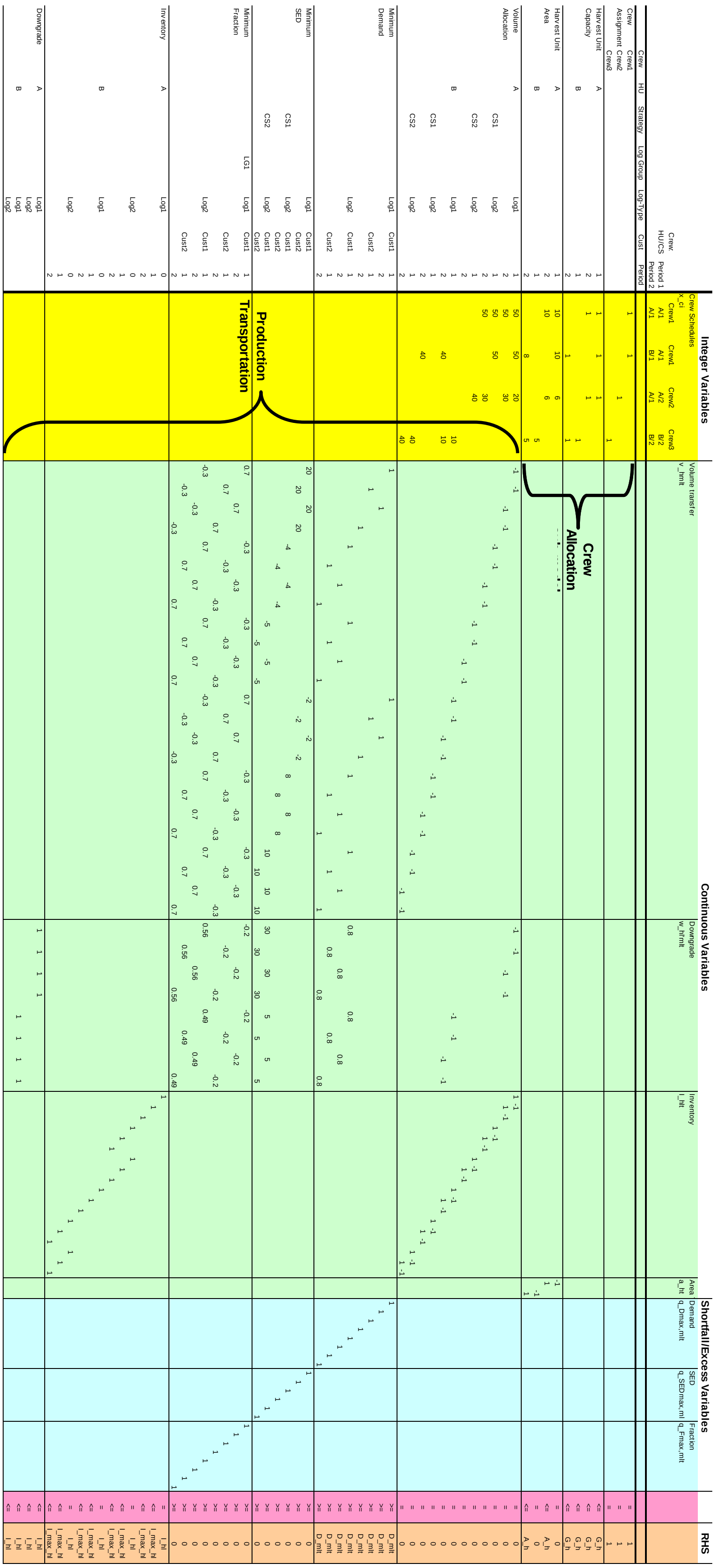


Figure 6.1: The matrix layout, excluding maximum demand, fraction and SED constraints

6.4 Crew Allocation constraints

These constraints represent the part of the OHS problem that is similar to a GSPP, with the addition of the harvest unit area constraints. These constraints govern crew allocation and state:

- a crew can only be allocated once;
- the number of crews operating in a harvest unit is restricted;
- the area harvested in a single harvest unit is restricted.

There is no need for a constraint on the cutting strategies as they represent instructions to the crews. Therefore, many crews are able to use identical cutting strategies.

6.4.1 Crew assignment

This constraint guarantees that each crew can appear only once in the solution. Without this constraint, the same crew could be placed in two or more different harvest units producing at full capacity. The form of this constraint is similar to GUB constraints found in the formulation of many scheduling problems (Section 5.1.4).

$$\sum_{i} x_{ci} = 1 \ \ldots \quad \forall c \tag{6.1}$$

where:

x_{ci}	is the allocation of crew c to schedule i;
c	indexes the crews;
i	indexes possible schedules.

This constraint is shown in Figure 6.2

		Crew Schedules x_ci					
Crew:		Crew1	Crew1	Crew2	Crew3		
HU/CS	Period 1	A/1	A/1	A/2	B/2		
	Period 2	A/1	B/1	A/1	B/2		
	Crew						
Crew	Crew1	1	1			=	1
Assignment	Crew2			1		=	1
	Crew3				1	=	1

Figure 6.2: The crew assignment constraint

6.4.2 Harvest unit constraints

6.4.2.1 Harvest unit integrality

As crews can only harvest in a single location at a time, crew location is governed by binary integer variables (x_{ci}). The integer requirement is forced on a solution through the branch and bound process (Section 5.2.3). However, in some cases the enforcement of a strict integer solution will give undesirable effects, especially when crew movement is considered. This aspect of the problem is discussed in detail in Section 7.1

6.4.2.2 Harvest unit compatibility

For the reasons given in Section 2.5.1, crews are not able to harvest all available harvest units. Instead, they can only harvest those units for which they are equipped. This compatibility information is provided as input data and is used to determine valid crew schedules. Therefore, a crew schedule that allocates a crew to an incompatible harvest unit will not appear in the problem, and will not be generated by column generation.

6.4.2.3 Harvest unit capacity

The parameter G_h limits the number of crews that can operate safely in harvest unit h. This figure is affected by a number of factors including the size of the landing, the accessibility of the harvest unit and the type of operations possible (see Section 2.5.1.4). Generally, only harvest units suitable for ground-based logging are able to support more than a single crew.

$$\sum_{c,i} H^i_{ht} x_{ci} \leq G_h \ \ldots \quad \forall h, t \tag{6.2}$$

where:

- G_h is the maximum number of crews allowed in harvest unit h;
- h indexes the harvest units;
- t indexes periods.

This constraint can be simplified if

$$G_h = 1 \ldots \forall h.$$

However, some of the case studies discussed in Chapter 9 allowed multiple crews to operate in a single harvest unit. Therefore, this simplification was not included.

Figure 6.3 shows this constraint. In the figure the first crew schedule places crew one in harvest unit **A** for both periods, the second crew schedule shows crew one in **A** in period one and **B** in period 2.

			Crew Schedules x_ci					
Crew:			Crew1	Crew1	Crew2	Crew3		
HU/CS	Period 1		A/1	A/1	A/2	B/2		
	Period 2		A/1	B/1	A/1	B/2		
	HU	Period						
Harvest Unit	A	1	1	1	1		<=	G_h
Capacity		2	1		1		<=	G_h
	B	1				1	<=	G_h
		2		1		1	<=	G_h

Figure 6.3: The harvest unit capacity constraint

6.4.2.4 Harvest unit area, a_{ht}

When a crew has harvested all the area of a harvest unit, it must move into another (see Section 6.4.2.1). The area of each harvest unit is A_h, and the area that a crew harvests within that harvest unit is determined by the crew productivity and the cutting strategy that the crew is using. The variable a_{ht} tracks the total cumulative area of harvest unit h that has been harvested up to period t.

$$\sum_{c,i} A^{ci}_{ht} x_{ci} + a_{h(t-1)} - a_{ht} \; = 0 \;\ldots\quad \forall h, t < |T|. \tag{6.3}$$

where:

A^{ci}_{ht} is the area of harvest unit h harvested by crew c following schedule i;

a_{ht} is the cumulative area harvested in harvest unit h by period t;

A_h is the total area in harvest unit h.

Equation (6.3) states that the area of harvest unit h harvested in period t $(\sum_{c,i} A^{ci}_{ht} x_{ci})$, plus the area of h harvested up to the end of t-1 $(a_{h(t-1)})$, is equal to the total area of h harvested up to the end of t (a_{ht}).

In the final period $|T|$ the area of the harvest unit A_h is included

$$\sum_{c,i} A^{ci}_{ht} x_{ci} + a_{h(t-1)} \; \leq A_h \;\ldots\quad \forall h, t = |T| \tag{6.4}$$

Equation (6.4) states that the area of harvest unit h harvested in the final period ($\sum_{c,i} A^{ci}_{ht} x_{ci}$), plus the area of h harvested up to the end t-1 ($a_{h(t-1)}$), is less than or equal to the total area of h (A_h).

Note, that constraint Equation (6.3) is formulated on a period-by-period basis. By formulating the constraint in this manner, the period where Equation (6.4) is active may be changed to force a harvest unit to be finished by a chosen period within the time horizon.

In Figure 6.4, the area constraints are shown for two periods giving a single area linking variable (a_{ht}) for each harvest unit.

			Crew Schedules x_ci				Area Transfer a_ht			
Crew:			Crew1	Crew1	Crew2	Crew3				
HU/CS	Period 1		A/1	A/1	A/2	B/2				
	Period 2		A/1	B/1	A/1	B/2				
	HU	Period								
Harvest Unit	A	1	10	10	6		-1		=	0
Area		2	10		6		1		<=	A_h
	B	1				5		-1	=	0
		2		8		5		1	<=	A_h

Figure 6.4: The harvest unit area constraint

6.4.2.5 Harvest unit completion

A crew may be forced to complete a harvest unit once it has entered the unit. The following equation indicates that a_{ht} will either equal zero or A_h in a feasible solution.

$$a_{ht} = \{0, A_h\} \ldots \forall h, t = T \tag{6.5}$$

where:

a_{ht} is the cumulative area harvested in harvest unit h by period t;
A_h is the total area in harvest unit h.

For harvest units with high entry costs, e.g., hauler units, it is too expensive to have a crew complete the unit in two separate operations (see Section 3.4.2). Unfortunately, when the time horizon is short the cost of returning to the harvest unit is not considered, a short term solution may move a crew out of a harvest unit and not consider the cost of moving the crew back. Therefore, Equation (6.5) may be required for some harvest units.

Equation (6.5) restricts the movement of crews away from a harvest unit before it is completed. It is enforced in the branch and bound algorithm. However, since the harvest units with this constraint usually only allow a single crew to harvest at a time ($G_h = 1$), valid variables in the integer solution will have additional structure (see Section 7.2.2). If only variables with this structure are generated the RLP solution will approximate the integer solution better.

Similarly, for reasons discussed in Section 3.4.2 a good OHS solution will not allow crews to leave small areas unharvested. In the branch and bound algorithm (see Section 8.3.2.2) a crew is not allowed to move if it would leave behind a small unharvested area. The crew is forced to remain and finish the harvest unit.

6.4.3 Cutting strategies

A *Cutting strategy* is a subset of log-types together with bucking instructions (i.e., the relative prices for a DP bucker[2]). Cutting strategies, together with harvest unit information (inventory assessments) are used to generate yield predictions.

The total number of log-types harvested at any one time by a crew is limited for reasons discussed in Section 2.5.1.4. This restriction is included implicitly when cutting strategies limit the number of log-types produced. As the number of log-types in a single strategy is limited, restricting a crew to a single strategy in a period will limit the number of log-types harvested.

A feasible OHS solution will limit the number and type of cutting strategies used in a period. These restrictions are similar to the harvest unit integrality constraints (Section 6.4.2.1), and are enforced in the branch and bound algorithm (see Section 8.3.1.2).

Cutting strategies contain a subset of logs. Two cutting strategies can share the same subset of log-types, but contain different bucking instructions. Combining these two strategies on a skid site will be less difficult than using completely different sets of log-types. Cutting strategies that share the same log-types are known as *Complementary strategies*. The set Q_k contains the complementary strategies indexed by k.

One of the following three statements can be used to control the number of cutting strategies used by a crew at any one time.

- Allow a crew to use any number of cutting strategies within a period.
- Only allow a single strategy per period (however, two strategies are allowed if the crew moves between harvest units).

$$\sum_i S^i_{st} x_{ci} = \{0, 1\} \ldots \forall c, s, t$$

- Allow multiple complementary strategies per period.

$$\sum_{i,s \in Q_k} S^i_{st} x_{ci} = \{0, 1\} \ldots \forall c, t, k$$

[2]Priority list buckers would need a list of priorities, while some systems may alter log specifications slightly to change volumes (Laroze 1999).

where:

S^i_{st} $= 1$ when schedule i includes strategy s in period t and zero otherwise;

Q_k is the set of complementary strategies;

k indexes the sets of complementary cutting strategies.

6.5 Production/Transportation constraints

The *Production/ Transportation (*PT*) constraints* (see Section 6.1.1) can make the OHS problem difficult to solve, as they tend to produce fractional solutions. This push towards fractional solutions occurs when PT constraints form extreme points away from integer solutions to the CA sub-model. If for instance a crew does not supply the correct mix of log-types to the market, the PT constraints may try to divide the crew among several harvest units. This division causes fractions to appear in the solution, forcing it away from integer solutions to the CA sub-model. The inclusion of elastic constraints in the modelling of Demand, Product properties and Product Groups constraints (Sections 6.5.2–6.5.4) may mitigate this effect.

The allocation of volume from the harvest units to the customers (with log-stocks and downgrading) is decided within the PT sub-model. This sub-model is essentially a resource constrained transportation problem that depends on:

- the production of the crews;
- the demands of the customers;
- the transportation cost from harvest unit to customer.

There is mutual interaction between the sub-models. The crews' production depends on the crew allocation. In addition, the costs within the transportation problem are important factors in the crew allocations because crews should be placed near customers.

6.5.1 Volume allocation, v_{hmlt}

The volume allocation constraint forms the basis of a transportation problem. Volume produced by the crews (P^{ci}_{hlt}) is allocated to the customers by the v_{hsmlt} variables. Some complications to this relationship are:

- the ability to maintain log-stocks between the periods;
- the ability to downgrade production from other log-types to satisfy a customer;
- the requirements to maintain average log properties and log group fractions.

The following volume allocation and market constraints are likely to change between different forestry companies. Possible restrictions that may be used by any company are discussed in Section 3.5. Therefore, the constraints given here are only an example of possible constraints. Where applicable, some possibilities for other formulations are mentioned.

The volume allocation constraint Equation (6.6) controls the volume produced, held in log-stocks and sold.

$$\begin{aligned} &\sum_{c,i} P^{ci}_{hlt} x_{ci} - \sum_{l' \neq l, m} (w_{hmll't}) \\ &+ (y_{hl(t-1)} - y_{hlt}) - \sum_{m} v_{hmlt} = 0 \qquad \ldots \forall h, l \notin \Xi, t \end{aligned} \tag{6.6}$$

where:

m	indexes the customers;
v_{hmlt}	is the volume allocation of log-type l in harvest unit h to customer m;
$w_{hmll't}$	is the volume of log-type l in harvest unit h downgraded to log-type l' for customer m in period t;
y_{hlt}	is the log-stocks of log-type l in harvest unit h at the end of period t;
Ξ	is the set of logs with SED requirements.

Equation (6.6) states that the total production of log-type l in harvest unit h in period t ($\sum_{c,i} P^{ci}_{hlt} x_{ci}$), minus the volume removed from this log-type by downgrading ($\sum_{l' \neq l,m} (w_{hmll't})$), plus the inventory from the previous period minus the inventory left at the end of the current period ($y_{hl(t-1)} - y_{hlt}$), is equal to the volume transferred to customers ($\sum_m v_{hmlt}$).

When log-types have active log property constraints, (see Section 6.5.3) they must be treated differently. The log-types with these constraints are elements of the Ξ set. For log-type l and $l \in \Xi$ the form of Equation (6.6) must be changed because the cutting-strategy (s) must be recorded against the volume. Therefore, the variable v_{hsmlt} is substituted for v_{hmlt} in Equation (6.7).

Similarly, the inventory is recorded by strategy y_{hslt} and downgrades $w_{hsml'l,t}$, $w_{hsmll't}$. These modifications significantly increase the number of constraints in the formulation. Therefore, they are only implemented when necessary.

$$\begin{aligned}&\sum_{c,i} P^{ci}_{hslt} x_{ci} - \sum_{l' \neq l, m} (w_{hsmll't}) \\ &+(y_{hsl(t-1)} - y_{hslt}) - \sum_{m} v_{hsmlt} = 0 \qquad \ldots \forall h, s, l \in \Xi, t.\end{aligned} \tag{6.7}$$

For the logs in Ξ, we then make the appropriate summations for ease of reference

$$\begin{aligned} v_{hmlt} &= \sum_{s} v_{hsmlt} & \ldots\ & \forall h, l \in \Xi, t \\ y_{hlt} &= \sum_{s} y_{hslt} & \ldots\ & \forall h, l \in \Xi, t \\ w_{hmll't} &= \sum_{s} w_{hsmll't} & \ldots\ & \forall h, l \in \Xi, m, l' \neq l, t \end{aligned}$$

					Crew Schedules x_ci				Volume transfer v_hmlt	Downgrade w_hl'mlt	Inventory I_hlt		
					Crew1	Crew1	Crew2	Crew3					
				Period 1	A/1	A/1	A/2	B/2					
				Period 2	A/1	B/1	A/1	B/2					
	HU	Strategy	Log-Type	Period									
Volume Allocation	A		Log1	1	50	50	20		-1 -1	-1 -1	1 -1	=	0
				2	50		30		-1 -1	-1 -1	1 -1	=	0
		CS1	Log2	1	50	50			-1 -1		1 -1	=	0
				2	50		30		-1 -1		1 -1	=	0
		CS2	Log2	1			40		-1 -1		1 -1	=	0
				2					-1 -1		1 -1	=	0
	B		Log1	1				10	-1 -1	-1 -1	1 -1	=	0
				2		40		10	-1 -1	-1 -1	1 -1	=	0
		CS1	Log2	1					-1 -1		1 -1	=	0
				2		40			-1 -1		1 -1	=	0
		CS2	Log2	1				40	-1 -1		1 -1	=	0
				2				40	-1 -1		1 -1	=	0

Figure 6.5: The volume allocation constraint

The volume allocation constraint is shown in Figure 6.5. Only log-type two will have active SED constraints and is in Ξ. Therefore, the constraints for log-type two are repeated for each of the two cutting strategies.

In this example, the log volumes differ depending on the harvest unit and cutting strategy. There are two volume transfer variables (v_{hmlt}) for each log-type because there are two customers. The downgrade variables ($w_{hml'lt}$) allow only downgrades from log-type one to log-type two.

The volume transfer and downgrade variables will link into the demand constraints (Section 6.5.2) while the inventory variables (y_{hlt}) transfer volume between periods.

6.5.2 Demands, D_{mlt}^{max},D_{mlt}^{min}

A discussion of the possible demand restrictions is found in Section 3.5.1. Equations (6.8) and (6.9) constrain the log volume delivered to the customers to lie between D_{mlt}^{max} and D_{mlt}^{min} in each period.

Other possible demand constraints could include:

- a cumulative constraint on volume;
- an average constraint over several periods or the time horizon;
- a constraint restricting the difference in production from one period to the next.

The downgrade $w_{hmll't}$ is considered in Equations (6.8) and (6.9) as logs can be downgraded from one log-type to another for delivery to a customer.

$$\sum_{h} v_{hmlt} + \sum_{h,l' \neq l} W_{l'l} w_{hml'lt} - q_{mlt}^{Dmax} \leq D_{mlt}^{max} \; \ldots \quad \forall m, l, t \tag{6.8}$$

$$\sum_{h} v_{hmlt} + \sum_{h,l' \neq l} W_{l'l} w_{hml'lt} + q_{mlt}^{Dmin} \geq D_{mlt}^{min} \; \ldots \quad \forall m, l, t \tag{6.9}$$

where:

$W_{l'l}$	is the loss in volume when log-type l' is downgraded to l;
D_{mlt}^{max}	is the maximum demand for log-type l in period t by customer m;
D_{mlt}^{min}	is the minimum demand for log-type l in period t by customer m;
q_{mlt}^{Dmax}	is the volume of log-type l in period t in excess of the demand of customer m;
q_{mlt}^{Dmin}	is the volume of log-type l in period t in shortfall of the demand of customer m.

Equation (6.8) states that the total volume of log-type l delivered to customer m in period t ($\sum_h v_{hmlt}$), plus the volume downgraded into l from other log-types ($\sum_{h,l' \neq l} W_{l'l} w_{hml'lt}$), minus the excess ($q_{mlt}^{Dmin}$), is less than the maximum demand of the customer (D_{mlt}^{min}).

The shortfall and excess volumes (q_{mlt}^{Dmin},q_{mlt}^{Dmax}) make these violations of the constraints possible. This property is useful because in practice it is often impossible to find a feasible OHS that meets all of the customers' demands. Therefore, a solution must be found that has excesses and/or shortfalls (see Section 3.5.1). The magnitude and choice of logs in shortfall or excess is controlled by the objective function contribution (C^{Dmin}_{mlt},C^{Dmax}_{mlt}) of the q^{Dmin}_{mlt} and q^{Dmax}_{mlt} variables. The costs C^{Dmin}_{mlt} and C^{Dmax}_{mlt} may be altered by the harvest scheduler until a suitable solution is found. In the case studies, these penalties were set to be a multiple of the market price of the log-type. The penalties should be high enough to force the optimisation to deliver lower priced log-types. If the penalties are too high the optimisation may generate expensive solutions that only insignificantly decrease demand violation.

Figure 6.6 shows the influence of the volume transfer, downgrade and penalty variables on the demand constraint. This example only includes a minimum demand constraint, for logs that are not in Ξ. A section of the volume allocation constraint is included to indicate how volume is transferred from this constraint.

	HU	Strategy	Log-Type	Cust	Period	Volume transfer v_hmlt																								Downgrade w_hl'mlt								Demand q_Dmax,mlt									
Volume Allocation	A		Log1		1	-1	-1																							-1	-1															=	0
					2			-1	-1																							-1	-1													=	0
		CS1	Log2		1					-1	-1																																			=	0
					2							-1	-1																																	=	0
		CS2	Log2		1									-1	-1																															=	0
					2											-1	-1																													=	0
	B		Log1		1													-1	-1															-1	-1											=	0
					2															-1	-1															-1	-1									=	0
		CS1	Log2		1																	-1	-1																							=	0
					2																			-1	-1																					=	0
		CS2	Log2		1																					-1	-1																			=	0
					2																							-1	-1																	=	0
Minimum Demand			Log1	Cust1	1	1												1																				1								>=	D_mlt
					2			1												1																			1							>=	D_mlt
				Cust2	1		1												1																					1						>=	D_mlt
					2				1												1																					1					>=	D_mlt
			Log2	Cust1	1					1				1								1				1				1				1								1				>=	D_mlt
					2							1				1								1				1				1				1							1			>=	D_mlt
				Cust2	1						1				1								1				1				1				1								1		>=	D_mlt
					2								1				1								1				1				1				1								1	>=	D_mlt

Figure 6.6: The minimum demand constraint

In Figure 6.6, volume is transferred from the volume allocation constraints into the demand constraints by the volume allocation variables v_{hmlt} and downgrade linking variables $w_{hmll't}$. Volume was supplied to the volume allocation constraints by the crew schedule variables x_{ci} (see Section 6.7) that are not shown.

6.5.3 Product properties, SED^{max}_{ml},SED^{min}_{ml}

In Section 2.6.3, several aggregate restrictions on log demands are discussed. In particular these restrictions are:

- average SED requirements for an order;
- specific proportions of log-types in a single order.

The average SED (SED_{hsl}) of a log-type is obtained from the yield predictions. The average SED changes depending on the cutting strategy. Similar constraints could be applied to other log properties such as density or stiffness.

The SED restriction is an average applied over a group logs. The grouping could be determined in a number of ways that depend on specific customer requirements. The grouping could be:

- effective over the entire planning horizon;
- effective over each period;
- effective on specific orders/customers.

In the formulation used in this thesis, the first option is chosen. Therefore, only one constraint is needed per customer, log-type combination.

$$\begin{aligned} &\sum_{h,s,t}(SED_{hsl} - SED_{ml}^{max})v_{hsmlt} \\ &+ \sum_{h,s,l'\neq l,t}(W_{hsl'l}^{SED} - SED_{ml}^{max})W_{l'l}w_{hsml'l,t} - q_{ml}^{SEDmax} \leq 0 \qquad \ldots\forall m,l \in \Xi \end{aligned} \tag{6.10}$$

$$\begin{aligned} &\sum_{h,s,t}(SED_{hsl} - SED_{ml}^{min})v_{hsmlt} \\ &+ \sum_{h,s,l'\neq l,t}(W_{hsl'l}^{SED} - SED_{ml}^{min})W_{l'l}w_{hsml'l,t} + q_{ml}^{SEDmin} \geq 0 \qquad \ldots\forall m,l \in \Xi \end{aligned} \tag{6.11}$$

where:

SED_{hsl}	is the SED of log-type l harvested from harvest unit h with strategy s;
$W^{SED}_{hsl'l}$	is the SED of log-type l' harvested with strategy s in harvest unit h downgraded to log-type l;
SED^{min}_{ml}	is the minimum average SED for log-type l allowed by customer m;
SED^{max}_{ml}	is the maximum average SED for log-type l allowed by customer m;
q^{SEDmin}_{ml}	is the value less than the maximum required SED for customer m and log-type l;
q^{SEDmax}_{ml}	is the value more than the maximum required SED for customer m and log-type l;

Equation (6.10) constrains the average SED. It uses the sum of the volume-weighted[3] differences of the actual SED and the required average $(\sum_{h,s,t}(SED_{hsl} - SED^{max}_{ml})v_{hsmlt})$. The downgraded volume $w_{hml'lt}$ is accounted for separately $(\sum_{h,s,l'\neq l,t}(W^{SED}_{hsl'l} - SED^{min}_{ml})W_{l'l}w_{hsml'l,t})$. The parameter $W^{SED}_{hsl'l}$ is the expected SED of the downgraded logs. The SED of downgraded logs $W^{SED}_{hsl'l}$ will be different to SED_{hsl} because the logs were originally made to a different specification.

Figure 6.7 shows the operation of the minimum average SED constraint. The example shown is simplified as the SED does not change depending on the cutting strategy used. The more complex constraints detailed above will have additional variables for each different cutting strategy used.

	HU	Strategy	Log-Type	Cust	Period	Volume transfer v_hmlt																							
Volume	A		Log1		1	-1	-1																						
Allocation					2			-1	-1																				
		CS1	Log2		1					-1	-1																		
					2							-1	-1																
		CS2	Log2		1									-1	-1														
					2											-1	-1												
	B		Log1		1													-1	-1										
					2															-1	-1								
		CS1	Log2		1																	-1	-1						
					2																			-1	-1				
		CS2	Log2		1																					-1	-1		
					2																							-1	-1
Minimum			Log1	Cust1		20		20										-2		-2									
SED				Cust2			20		20										-2		-2								
		CS1	Log2	Cust1						-4		-4										8		8					
				Cust2							-4		-4										8		8				
		CS2	Log2	Cust1										-5		-5										10		10	
				Cust2											-5		-5										10		10

	HU	Strategy	Log-Type	Cust	Period	Downgrade w_hl'mlt								SED q_SEDmax,ml							
Volume	A		Log1		1	-1	-1													=	0
Allocation					2			-1	-1											=	0
		CS1	Log2		1															=	0
					2															=	0
		CS2	Log2		1															=	0
					2															=	0
	B		Log1		1					-1	-1									=	0
					2							-1	-1							=	0
		CS1	Log2		1															=	0
					2															=	0
		CS2	Log2		1															=	0
					2															=	0
Minimum			Log1	Cust1										1						>=	0
SED				Cust2											1					>=	0
		CS1	Log2	Cust1												1				>=	0
				Cust2													1			>=	0
		CS2	Log2	Cust1		30		30		5		5						1		>=	0
				Cust2			30		30		5		5						1	>=	0

Figure 6.7: The SED constraint

The SED constraint operates in a similar manner to the demand constraint. Note that the same variables that transfer volume to the demand constraint also transfer volume to the SED constraint.

[3]Normal practice is to use volume weighting to find Small End Diameter (SED) rather than number of logs.

6.5.4 Product fractions, F_{ml}^{max},F_{ml}^{min}

Product fraction restrictions are used for export log groups in particular (see Section 2.6.2). These constraints limit the volume of a particular log-type to a fraction of the total volume of a group of log-types. For instance, Japanese JM logs are restricted to be only 30% of the total volume of an order of JL and JM logs (from Table 2.1 in Section 2.6.3).

As with average SED restrictions, the product fraction constraints apply over a number of logs. In contrast to the average SED constraint, the product fraction constraints in this specific formulation apply in each period. Therefore, there is a constraint for each log group and customer in each period.

$$\begin{aligned}\sum_h v_{hmlt} + \sum_{h,l'\neq l} W_{l'l} w_{hml'lt} - F_{ml}^{min} \sum_{h,l\in G_k} v_{hmlt} \\ -F_{ml}^{min} \sum_{h,l\in G_k,l'\neq l} W_{l'l} w_{hml'lt} + q_{mlt}^{Fmin} \geq 0 \qquad \ldots \forall k,m,l \in G_k, t \end{aligned} \tag{6.12}$$

$$\begin{aligned}\sum_h v_{hmlt} + \sum_{h,l'\neq l} W_{l'l} w_{hml'lt} - F_{ml}^{max} \sum_{h,l\in G_k} v_{hmlt} \\ -F_{ml}^{max} \sum_{h,l\in G_k,l'\neq l} W_{l'l} w_{hml'lt} - q_{mlt}^{Fmax} \leq 0 \qquad \ldots \forall k,m,l \in G_k, t \end{aligned} \tag{6.13}$$

where:

F_{ml}^{max}	is the maximum fraction for log-type l for customer m;
F_{ml}^{min}	is the minimum fraction for log-type l for customer m;
q_{mlt}^{Fmax}	is the fraction of log-type l exceeding the maximum fraction for customer m in period t;
q_{mlt}^{Fmin}	is the fraction of log-type l less than the minimum fraction for customer m in period t.

Equation (6.12) states that the volume of log-type l delivered to customer m in period t ($\sum_h v_{hmlt}$), plus the volume of downgraded logs ($\sum_{h,l'\neq l} W_{l'l} w_{hml'lt}$), plus the violation of this constraint (q_{mlt}^{Fmin}), must be greater than the total volume of delivered logs in the log group G_k multiplied by the required minimum fraction $(F_{ml}^{min} \sum_{h,l \in G_k} v_{hmlt})$, plus the total volume of downgraded logs in the log group G_k multiplied by the required minimum fraction $(F_{ml}^{min} \sum_{h,l \in G_k, l' \neq l} W_{l'l} w_{hml'lt})$.

	HU	Strategy	Log Group	Log-Type	Cust	Period	Volume transfer v_hmlt																								Downgrade w_hl'mlt								Fraction q_Fmax,mlt									
Volume	A			Log1		1	-1	-1																							-1	-1															=	0
Allocation						2			-1	-1																							-1	-1													=	0
		CS1		Log2		1					-1	-1																																			=	0
						2							-1	-1																																	=	0
		CS2		Log2		1									-1	-1																															=	0
						2											-1	-1																													=	0
	B			Log1		1													-1	-1															-1	-1											=	0
						2															-1	-1															-1	-1									=	0
		CS1		Log2		1																	-1	-1																							=	0
						2																			-1	-1																					=	0
		CS2		Log2		1																					-1	-1																			=	0
						2																							-1	-1																	=	0
Minimum			LG1	Log1	Cust1	1	0.7				-0.3				-0.3				0.7				-0.3				-0.3				-0.2				-0.2				1								>=	0
Fraction						2			0.7				-0.3				-0.3				0.7				-0.3				-0.3				-0.2				-0.2			1							>=	0
					Cust2	1		0.7				-0.3				-0.3				0.7				-0.3				-0.3				-0.2				-0.2				1						>=	0
						2				0.7				-0.3				-0.3				0.7				-0.3				-0.3				-0.2				-0.2				1					>=	0
				Log2	Cust1	1	-0.3				0.7				0.7				-0.3				0.7				0.7				0.56				0.49								1				>=	0
						2			-0.3				0.7				0.7				-0.3				0.7				0.7				0.56				0.49								1			>=	0
					Cust2	1		-0.3				0.7				0.7				-0.3				0.7				0.7				0.56				0.49								1		>=	0
						2				-0.3				0.7				0.7				-0.3				0.7				0.7				0.56				0.49								1	>=	0

Figure 6.8: The log fraction constraint

The operation of the product fraction constraints is similar to the demand and SED constraints. In Figure 6.8, both logs in the example belong to a single group. Thus, the volume transfer and downgrade variables for each log appear in all fraction constraints. It should be noted that in the example the minimum fraction for both logs is 30%

6.5.5 Inventory, y_{hlt}

This constraint restricts the log-stocks within the model (see Section 3.5.2). The volume in the log-stocks (y_{hlt}) is transferred across the periods, so past production can meet future demands.

A number of possible restrictions can be placed on the levels on inventory:

- the total volume is restricted at the harvest units;
- the total volume is restricted across the harvest units;
- the time logs spend in log-stocks is limited (because of sap stain);
- the number of different log-types that can be stored at a landing is restricted.

The Equation (6.15) handles the first item, while the fourth is implicitly handled by restricting the number of log-types in the cutting strategies (Section 6.4.3). The second and third items are not treated in this thesis. The second item may be modelled trivially by another constraint while the third item is slightly more difficult to model.

Equation (6.14) sets the initial inventory for harvest units to the parameter I_{hl}.

$$y_{hlt} = I_{hl} \quad \ldots \quad \forall h, l, t = 0 \tag{6.14}$$

$$y_{hlt} \leq I_{hl}^{max} \quad \ldots \quad \forall h, l, t \tag{6.15}$$

where:

y_{hlt} is the log-stocks of log-type l in harvest unit h at the end of period t;

I_{hl}^{max} is the maximum log-stocks of log-type l held in harvest unit h;

I_{hl} is the initial log-stocks of log-type l in harvest unit h.

					Downgrade w_hl'mlt								Inventory I_hlt																			
	HU	Strategy	Log-Type	Period																												
Volume	A		Log1	1	-1	-1							1	-1																	=	0
Allocation				2			-1	-1						1	-1																=	0
		CS1	Log2	1												1	-1														=	0
				2													1	-1													=	0
		CS2	Log2	1															1	-1											=	0
				2																1	-1										=	0
	B		Log1	1					-1	-1												1	-1								=	0
				2							-1	-1											1	-1							=	0
		CS1	Log2	1																					1	-1					=	0
				2																						1	-1				=	0
		CS2	Log2	1																								1	-1		=	0
				2																									1	-1	=	0
Inventory	A		Log1	0									1																		=	I_hl
				1										1																	<=	I_max_hl
				2											1																<=	I_max_hl
			Log2	0												1			1												=	I_hl
				1													1			1											<=	I_max_hl
				2														1			1										<=	I_max_hl
	B		Log1	0																		1									=	I_hl
				1																			1								<=	I_max_hl
				2																				1							<=	I_max_hl
			Log2	0																					1			1			=	I_hl
				1																						1			1		<=	I_max_hl
				2																							1			1	<=	I_max_hl
Downgrade	A		Log1		1	1	1	1																							<=	I_hl
			Log2																												<=	I_hl
	B		Log1						1	1	1	1																			<=	I_hl
			Log2																												<=	I_hl

Figure 6.9: The inventory and downgrade constraints

Figure 6.9 shows the operation of the inventory constraint.

6.5.6 Downgrading control, $w_{hml'lt}$

Downgrading occurs when the volume produced in one log-type is sold as another log-type (see Section 3.5.4). This downgrading is complementary to the re-grading that occurs when the relative values are adjusted to generate the yield predictions (see Section 3.3.1). The dimensionless conversion factor $W_{l'l}$ is used to calculate the volume lost downgrading. Volume may be lost because sections of the log may be removed at the landing to make it suitable for the new log-type.

The volume of downgrade from a harvest unit h and log-type l to a customer m and log-type l' is tracked by the variable $w_{hmll't}$. This level of detail allows control over which customers are willing to accept downgrades for their logs. Calculation of the log property requirements also tracks customer and harvest unit.

To limit downgrading to volumes that already exist in stockpiles (as discussed in Section 3.5.4) the optional constraint Equation (6.16) may be used.

$$\sum_{l' \neq l, t} w_{hmll't} \leq I_{hl} \; \ldots \quad \forall h, l \tag{6.16}$$

where:

$w_{hmll't}$	is the volume of log-type l in harvest unit h downgraded to log-type l' for customer m in period t
I_{hl}	is the initial log-stocks of log-type l in harvest unit h.

This is an optional constraint. Its operation is shown in Figure 6.9. Note, the constraint limits the volume that is downgraded, not the volume produced from downgrades (which would be multiplied by $W_{l'l}$).

6.6 Objective

The objective in this formulation is to maximise the net value of the logs sold. This value is the revenue from delivered logs, minus the harvesting and transportation costs. The transportation costs are specific to the harvest unit, as the distances travelled from the harvest unit to the customer can be significant. However, in this formulation, a minimise objective function is used. The value minimised is simply the profit multiplied by -1.

6.6.1 Log revenue, $c_{P,ml}$

Revenue from logs is only accrued when they are sold. There are a number of different methods used to price logs, from stumpage sales through to delivered sales (see Section 2.6.4). These methods allocate the costs of production and harvesting between the customer and supplier in different ways. The optimisation model must, however, use a common reference point to evaluate the total profit. All log sales are therefore priced using the delivered price ($c_{P,ml}$, \$/m^3). Because the delivered price is used, logs sourced from different parts of the forest and delivered to the same customer all generate the same revenue. The transport cost can be considered separately and the most efficient allocation of the logs to customers that minimises the cost of transportation may be found.

If stumpage prices were used, the customer will pay more for logs (they pay transportation) if they are harvested farther from the customer's location. If the forestry company makes a profit on selling transportation, the transportation problem would be distorted and the optimisation would attempt to *maximise* transportation costs. The optimisation will therefore give inefficient solutions to the OHS problem.

6.6.2 Transportation costs, $c_{T,hml}$

Transportation costs reflect the cost of transporting logs from the landing to a customer. These costs can be based on distance, type of truck used, and the length of the logs. An individual cost ($c_{T,hml}$, \$/m^3) is applied to each harvest unit, log-type, and customer combination.

6.6.3 Penalty costs, C_{mlt}^{Dmax},C_{ml}^{SEDmax},C_{mlt}^{Fmax}

These costs control the variables (q_{mlt}^{Dmax},q_{ml}^{SEDmax},q_{mlt}^{Fmax}) that make the market constraints 'soft'. The cost will reduce the violation of the

- Demand constraints (C_{mlt}^{Dmax},C_{mlt}^{Dmin});
- SED constraints (C_{ml}^{SEDmax},C_{ml}^{SEDmin});
- Product fraction constraints (C_{mlt}^{Fmax},C_{mlt}^{Fmin});

Higher penalties should result in lower violations if feasible. In practice, these costs should be related to the actual costs of not meeting the restrictions. For example, the cost (C_{mlt}^{Dmin}) of shortfall in the demand restriction should be set to the cost of obtaining the logs from an outside supplier to meet demand.

The following two sections Sections 6.6.4 and 6.6.5 discuss objective value coefficients that are used to control some end effects of the model. A detailed discussion of end-effects is found in Section 8.6.1.

6.6.4 Residual harvest unit value, $c_{R,h}$

Downgrading on the landing and re-grading while logmaking make it possible for high-value harvest units to be felled in order to satisfy demand for low-value logs. While satisfying this low-value demand may be the best decision in the short-term, the use of these harvest units will reduce the ability to meet high quality demand in periods beyond the end of the time horizon. To counter this tendency for short-term gain at the expense of long-term production, a residual cost ($c_{R,h}$, \$/h) of the harvest units is included.

The residual cost reflects the value to a long-term solution of having the area available at the end of the short-term planning horizon. To calculate this cost accurately the dual information from a tactical planning solution can be used. However, if information from the tactical plan is unavailable, the value of the timber in the harvest unit if it was sold in an unrestricted market can be used. This value will be multiplied by a factor that represents the expected loss in value when the market is restricted by actual customer demands.

6.6.5 Residual inventory value, $c_{IR,hl}$

Any revenue from sales of log-stocks remaining at the end of the time horizon will not be considered in this model. If logs are downgraded to lower value log-types and sold within the time horizon the revenue is available to the model. however, the long-term consequences for revenue is worse, as these high-value logs are wasted. To prevent this behaviour, a residual value ($c_{IR,hl}$, \$/m^3) is placed on the volume of log-stocks so that log-stocks are kept past the end of the time horizon.

6.6.6 Objective equation

The complete formulation of the objective function to be minimised is,

$$
\begin{aligned}
z = & \sum_{c,i} c_{ci} x_{ci} \\
& - \sum_{h,m,l,t} (c_{P,ml} - c_{T,hml}) v_{hmlt} \\
& - \sum_{h,m,l,t} (c_{P,ml} - c_{T,hml}) w_{hml'lt} \\
& + \sum_{m,l,t} C^{Dmax}_{mlt} q^{Dmax}_{mlt} + \sum_{m,l,t} C^{Dmin}_{mlt} q^{Dmin}_{mlt} \\
& + \sum_{m,l} C^{SEDmax}_{ml} q^{SEDmax}_{ml} + \sum_{m,l} C^{SEDmin}_{ml} q^{SEDmin}_{ml} \\
& + \sum_{m,l,t} C^{Fmax}_{mlt} q^{Fmax}_{mlt} + \sum_{m,l,t} C^{Fmin}_{mlt} q^{Fmin}_{mlt} \\
& + \sum_{h,t=T} c_{R,h}(A_h - a_{ht}) + \sum_{h,l,t=T} c_{IR,hl} y_{hlt}
\end{aligned}
\tag{6.17}
$$

where:

c_{ci} is the cost of crew c following schedule i;

$c_{P,ml}$ is the revenue from log-type l delivered to customer m;

$c_{T,hml}$ is the transportation cost to move log l from harvest unit h to customer m;

$c_{R,h}$ is the value of the residual area of harvest unit h;

$c_{IR,hl}$ is the value of the residual log-stocks of log-type l left in harvest unit h at the end of the time horizon.

C^{Dmin}_{mlt}	is the cost associated with shortfall volume of log-type l in period t for customer m;
C^{Dmax}_{mlt}	is the cost associated with excess volume of log-type l in period t for customer m ;
C^{SEDmin}_{ml}	is the cost of violating the SED minimum for customer m and log-type l;
C^{SEDmax}_{ml}	is the cost of violating the SED maximum for customer m and log-type l;
C^{Fmin}_{mlt}	is the cost of the penalty variables on the minimum log fraction constraints;
C^{Fmax}_{mlt}	is the cost of the penalty variables on the maximum log fraction constraints;

6.7 Crew schedule variables, x_{ci}

The previous discussion focused on the constraints of the model. However, in the OHSA the construction of the crew-schedules is very important.

A x_{ci} variable is internally represented by the combination of a crew with a schedule. The information in the resulting crew schedule is used directly in the crew constraints and in the harvest unit capacity constraints.

The x_{ci} variables control crew allocations and the area harvested, however, the x_{ci} variables affect the PT constraints of the problem. These effects stem from the volume of logs produced by a crew following the schedule.

The area (A^{ci}_{ht}) harvested in harvest unit h in period t is calculated. In addition the production (P^{ci}_{hlt}) of log-type l in harvest unit h and period t is found. These figures are placed in the appropriate rows in the column that represents the crew schedule. The derivation of these two figures is described in Sections 6.7.2 and 6.7.3

The crew schedule variables influence the behaviour of the PT sub-model. The schedule determines where the crew harvests and the cutting strategies used throughout the planning horizon. The volume and properties of logs harvested can be calculated from the crew schedule along with the area of each harvest unit felled, and the total cost of harvesting.

In contrast with a Model I formulation, this Model II formulation can easily include information from previous periods when calculating the values above. Thus, production volume and the cost of production in a period can directly refer to the previous location of the crew.

6.7.1 Movement penalties, $M^{\$}_{h'h}$,$M^{P}_{h'h}$

In a Model II formulation the movement costs and penalties do not appear in the formulation explicitly, instead they are used to calculate the coefficients of the x_{ci} variables.

When a crew moves from one harvest unit h' to another h there can be a monetary cost and a loss of production associated with this move (Section 3.4.2). The movement cost ($M^{\$}_{h'h}$) is the cost in dollars to move from h to h'. This cost must be added to the overall costs of operating the crew. If the crew does not move

$$\begin{aligned} & H^i_{h(t-1)} = H^i_{ht} = 1 \\ \implies & H(i,t-1) = H(i,t) = h' = h \\ \implies & M^{\$}_{h'h} = 0. \end{aligned}$$

The production penalty ($M^{P}_{h'h}$) is the reduced productivity of any crew when they move between h' and h. The factor $M^{P}_{h'h}$ lies between zero and one ($0 \leq M^{P}_{h'h} \leq 1$) and multiplies the crew production (P_{ch}) when a crew moves. Therefore, if a crew will lose one day out of five when it moves between the harvest units the production penalty is $M^{P}_{h'h} = 0.8$. If a crew does not move

$$\begin{aligned} & H^i_{h(t-1)} = H^i_{ht} = 1 \\ \implies & H(i,t-1) = H(i,t) = h' = h \\ \implies & M^{P}_{h'h} = 1. \end{aligned}$$

As, the cost and the loss of productivity can vary based on the distance between the harvest units and on other factors, $M^{\$}_{h'h}$ and $M^{P}_{h'h}$ are allowed to vary based on the origin harvest unit h' and the destination h. The costs and penalties of movement are input parameters for the model.

6.7.2 Crew area, A_{ht}^{ci}

Yield predictions give the predicted total volume per hectare. The reciprocal of this figure is $A_{Y,hs}$ (hectares/m^3). The area coefficient (A_{ht}^{ci}) for the crew schedule is derived as shown below.

$$A_{ht}^{ci} = H_{ht}^{i} P_{ch} M_{h'h}^{P} A_{Y,hs} \quad h' = H(i, t-1), s = S(i,t) \ldots \forall c, i, h, t \tag{6.18}$$

where:

A_{ht}^{ci} is the area of harvest unit h harvested by crew c following schedule i;

$A_{Y,hs}$ is the hectares of h cleared per m^3 harvested using strategy s.

6.7.3 Crew production, P_{hlt}^{ci}

The yield prediction information for each harvest unit is obtained from the inventory system (Section 2.4.5). The data are converted to a fraction ($P_{F,hsl}$) of each log-type that is produced by the cutting strategy in this harvest unit. This fraction is converted to a volume production rate when multiplied by the crew's productivity (P_{ch}).

The productivity of a given crew is determined by the crew's size, equipment and the conditions under which they operate (Section 3.4.1). The crew productivity (P_{ch}) in m^3 per day depends on the crew and the harvest unit they harvest.

$$P_{hlt}^{ci} = H_{ht}^{i} P_{ch} M_{h'h}^{P} P_{F,hsl} \quad h' = H(i, t-1), s = S(i,t) \ldots \forall c, i, h, l \notin \Xi, t \tag{6.19}$$

where:

P_{hlt}^{ci} is the production of log-type l by crew c following schedule i;

Ξ is the set of logs with SED requirements.

The production coefficient (P_{hlt}^{ci}), does not track which strategy was used. For log properties (for instance SED) that are dependent on the strategy, the strategy used becomes important. For logs that require constraints on these log properties, $l \in \Xi$, (see Section 6.5.3) the coefficient P_{hslt}^{ci} is given by

$$P^{ci}_{hslt} = H^i_{ht} P_{ch} P_{F,hsl} M^P_{h'h} S^i_{st} \quad h' = H(i,t-1) \ldots \forall c,i,h,s,l \in \Xi, t. \tag{6.20}$$

6.7.4 Crew schedule cost, c_{ci}

The cost (c_{ci} \$/m^3) of a crew schedule variable (x_{ci}) is the sum of the harvesting and movement costs of crew c carrying out the schedule i. Each crew is allocated a production cost (C_{ch}) to harvest harvest unit h. This cost depends on the crew's equipment and the type of harvest unit (Section 3.4.1). Multiplying this figure by the productivity gives the crew cost for that harvest unit. Any movement costs of the schedule are then added.

$$c_{ci} = \sum_{\substack{t \\ h=H(i,t) \\ h'=H(i,t-1)}} P_{ch} M^P_{h'h} C_{ch} + \sum_{\substack{t \\ h=H(i,t) \\ h'=H(i,t-1)}} M^{\$}_{h'h} \ldots \forall c, i \tag{6.21}$$

where:

c_{ci} is the cost of crew c following schedule i.

6.7.5 Crew schedule variable structure

Once the various coefficients (H^i_{ht},A^{ci}_{ht},P^{ci}_{hlt}) are defined, the column that represents a particular crew schedule in the constraint matrix ($\boldsymbol{a}_{ci}$) can be constructed as follows.

This column can be partitioned into five parts:

- the crew assignment constraints denoted $\boldsymbol{a}^C_{ci}$;
- the harvest unit capacity constraints denoted $\boldsymbol{a}^H_{ci}$;
- the harvest unit area constraints denoted $\boldsymbol{a}^A_{ci}$;
- the volume allocation constraints denoted $\boldsymbol{a}^P_{ci}$;
- the rest of the constraints denoted $\boldsymbol{a}^N_{ci}$.

$$\boldsymbol{a}_{ci} = \begin{pmatrix} \boldsymbol{a}_{ci}^{C} \\ \boldsymbol{a}_{ci}^{H} \\ \boldsymbol{a}_{ci}^{A} \\ \boldsymbol{a}_{ci}^{P} \\ \boldsymbol{a}_{ci}^{N} \end{pmatrix} \tag{6.22}$$

where:

$\boldsymbol{a}_{ci}$ is the column of the $\boldsymbol{A}$ matrix that corresponds to a x_{ci} variable.

It is easy to see that

$$\boldsymbol{a}_{ci}^{C} = \boldsymbol{e}_{c}$$

where:

e_i is the i^{th} column of the identity matrix.

The other partitions are obtained by simply placing the correct parameter values into the appropriate rows.

$$\begin{aligned} \boldsymbol{a}_{ci}^{H} =& [H_{1,1}^{i} \dots H_{1,|T|}^{i}, H_{2,1}^{i} \dots H_{2,|T|}^{i}, \dots H_{|H|,|T|}^{i}]^{T} \\ \boldsymbol{a}_{ci}^{A} =& [A_{1,1}^{ci} \dots A_{1,|T|}^{ci}, A_{2,1}^{ci} \dots A_{2,|T|}^{ci}, \dots A_{|H|,|T|}^{ci}]^{T} \\ \boldsymbol{a}_{ci}^{P} =& [P_{1,1,1}^{ci} \dots P_{1,1,|T|}^{ci} \dots P_{1,|L|,|T|}^{ci} \dots P_{|H|,|L|,|T|}^{ci}]^{T} \\ \boldsymbol{a}_{ci}^{N} =& \boldsymbol{0} \end{aligned} \tag{6.23}$$

where:

H_{ht}^{i} $= 1$ when schedule i contains harvest unit h in period t and zero otherwise;

A_{ht}^{ci} is the area of harvest unit h harvested by crew c following schedule i;

P_{hlt}^{ci} is the production of log-type l by crew c following schedule i.

Figure 6.10 shows four example crew schedules. In this figure, the area and the volumes produced change as the crews change harvest area and cutting strategy.

						Crew Schedules x_ci					
				Crew: HU/CS		Crew1	Crew1	Crew2	Crew3		
					Period 1	A/1	A/1	A/2	B/2		
					Period 2	A/1	B/1	A/1	B/2		
	Crew	HU	Strategy	Log-Type	Period						
Crew Assignment	Crew1					1	1			=	1
	Crew2							1		=	1
	Crew3								1	=	1
Harvest Unit Capacity		A			1	1	1	1		<=	G_h
					2	1		1		<=	G_h
		B			1				1	<=	G_h
					2		1		1	<=	G_h
Harvest Unit Area		A			1	10	10	6		=	0
					2	10		6		<=	A_h
		B			1				5	=	0
					2		8		5	<=	A_h
Volume Allocation		A		Log1	1	50	50	20		=	0
					2	50		30		=	0
			CS1	Log2	1	50	50			=	0
					2	50		30		=	0
			CS2	Log2	1			40		=	0
					2					=	0
		B		Log1	1				10	=	0
					2		40		10	=	0
			CS1	Log2	1					=	0
					2		40			=	0
			CS2	Log2	1				40	=	0
					2				40	=	0

Figure 6.10: Crew schedule variables in the constraint matrix

6.8 Concise formulation

$$
\begin{aligned}
\underset{x_{ci},\, v_{hmlt}}{\text{Minimise}} \quad z = & \sum_{c,i} c_{ci} x_{ci} \\
& - \sum_{h,m,l,t} (c_{P,ml} - c_{T,hml}) v_{hmlt} \\
& - \sum_{h,m,l,t} (c_{P,ml} - c_{T,hml}) w_{hml'lt} \\
& + \sum_{m,l,t} C^{Dmax}_{mlt} q^{Dmax}_{mlt} + \sum_{m,l,t} C^{Dmin}_{mlt} q^{Dmin}_{mlt} \\
& + \sum_{m,l} C^{SEDmax}_{ml} q^{SEDmax}_{ml} + \sum_{m,l} C^{SEDmin}_{ml} q^{SEDmin}_{ml} \\
& + \sum_{m,l,t} C^{Fmax}_{mlt} q^{Fmax}_{mlt} + \sum_{m,l,t} C^{Fmin}_{mlt} q^{Fmin}_{mlt} \\
& + \sum_{h,t=T} c_{R,h}(A_h - a_{ht}) + \sum_{h,l,t=T} c_{IR,hl} y_{hlt}
\end{aligned}
$$

subject to:

Crew Allocation

$$\sum_{i} x_{ci} = 1 \qquad \ldots \forall c$$

Harvest Unit Capacity

$$\sum_{c,i} H^{i}_{ht} x_{ci} \leq G_h \qquad \ldots \forall h, t$$

Harvest Unit Area

$$\sum_{c,i} A^{ci}_{ht} x_{ci} + a_{h(t-1)} - a_{ht} = 0 \qquad \ldots \forall h, t < |T|$$

$$\sum_{c,i} A^{ci}_{ht} x_{ci} + a_{h(t-1)} \leq A_h \qquad \ldots \forall h, t = |T|$$

Volume Allocation

$$\begin{aligned} &\sum_{c,i} P^{ci}_{hlt} x_{ci} - \sum_{l' \neq l, m} (w_{hmll't}) \\ &+(y_{hl(t-1)} - y_{hlt}) - \sum_{m} v_{hmlt} = 0 \qquad \ldots \forall h, l \notin \Xi, t \end{aligned}$$

Demand

$$\sum_{h} v_{hmlt} + \sum_{h, l' \neq l} W_{l'l} w_{hml'lt} - q^{Dmax}_{mlt} \leq D^{max}_{mlt} \qquad \ldots \forall m, l, t$$

$$\sum_{h} v_{hmlt} + \sum_{h, l' \neq l} W_{l'l} w_{hml'lt} + q^{Dmin}_{mlt} \geq D^{min}_{mlt} \qquad \ldots \forall m, l, t$$

SED

$$\sum_{h,s,t}(SED_{hsl} - SED_{ml}^{max})v_{hsmlt}$$
$$+\sum_{h,s,l'\neq l,t}(W_{hsl'l}^{SED} - SED_{ml}^{max})W_{l'l}w_{hsml'l,t} - q_{ml}^{SEDmax} \leq 0 \qquad \ldots\forall m, l \in \Xi$$

$$\sum_{h,s,t}(SED_{hsl} - SED_{ml}^{min})v_{hsmlt}$$
$$+\sum_{h,s,l'\neq l,t}(W_{hsl'l}^{SED} - SED_{ml}^{min})W_{l'l}w_{hsml'l,t} + q_{ml}^{SEDmin} \geq 0 \qquad \ldots\forall m, l \in \Xi$$

Fraction

$$\sum_{h} v_{hmlt} + \sum_{h,l'\neq l} W_{l'l}w_{hml'lt} - F_{ml}^{min}\sum_{h,l\in G_k} v_{hmlt}$$
$$-F_{ml}^{min}\sum_{h,l\in G_k,l'\neq l} W_{l'l}w_{hml'lt} + q_{mlt}^{Fmin} \geq 0 \qquad \ldots\forall k, m, l \in G_k, t$$

$$\sum_{h} v_{hmlt} + \sum_{h,l'\neq l} W_{l'l}w_{hml'lt} - F_{ml}^{max}\sum_{h,l\in G_k} v_{hmlt}$$
$$-F_{ml}^{max}\sum_{h,l\in G_k,l'\neq l} W_{l'l}w_{hml'lt} - q_{mlt}^{Fmax} \leq 0 \qquad \ldots\forall k, m, l \in G_k, t$$

Inventory

$$y_{hlt} = I_{hl} \qquad \ldots\forall h, l, t = 0$$
$$y_{hlt} \leq I_{hl}^{max} \qquad \ldots\forall h, l, t$$

Downgrade

$$\sum_{l'\neq l,t} w_{hmll't} \leq I_{hl} \qquad \ldots\forall h, l$$

and

$$
\begin{aligned}
x_{ci} &\in \{0, 1\} && \ldots \forall c, i \\
v_{hmlt} &\geq 0 && \ldots \forall h, m, l, t \\
y_{hlt} &\geq 0 && \ldots \forall h, l, t \\
w_{hml'lt} &\geq 0 && \ldots \forall h, l', l, t
\end{aligned}
$$

where:

- c indexes the crews;
- i indexes possible schedules;
- h indexes the harvest units;
- s indexes the possible cutting strategies;
- m indexes the customers;
- l indexes the log-types;
- k indexes the log-groups;
- t indexes periods;
- H_{ht}^{i} $= 1$ when schedule i contains harvest unit h in period t and zero otherwise;
- S_{st}^{i} $= 1$ when schedule i includes strategy s in period t and zero otherwise;
- x_{ci} is the allocation of crew c to schedule i;
- G_h is the maximum number of crews allowed in harvest unit h;
- a_{ht} is the cumulative area harvested in harvest unit h by period t;
- A_h is the total area in harvest unit h;
- P_{hlt}^{ci} is the production of log-type l by crew c following schedule i;
- A_{ht}^{ci} is the area of harvest unit h harvested by crew c following schedule i;
- v_{hmlt} is the volume allocation of log-type l in harvest unit h to customer m;
- y_{hlt} is the log-stocks of log-type l in harvest unit h at the end of period t;
- I_{hl}^{max} is the maximum log-stocks of log-type l held in harvest unit h;
- I_{hl} is the initial log-stocks of log-type l in harvest unit h;

$w_{hml'lt}$ is the volume of log-type l' in harvest unit h downgraded to log-type l for customer m in period t;

$w_{hmll't}$ is the volume of log-type l in harvest unit h downgraded to log-type l' for customer m in period t;

$W_{l'l}$ is the loss in volume when log-type l' is downgraded to l;

$W^{SED}_{hsl'l}$ is the SED of log-type l' harvested with strategy s in harvest unit h downgraded to log-type l;

D^{max}_{mlt} is the maximum demand for log-type l in period t by customer m;

D^{min}_{mlt} is the minimum demand for log-type l in period t by customer m;

q^{Dmax}_{mlt} is the volume of log-type l in period t in excess of the demand of customer m;

q^{Dmin}_{mlt} is the volume of log-type l in period t in shortfall of the demand of customer m;

SED_{hsl} is the SED of log-type l harvested from harvest unit h with strategy s;

SED^{min}_{ml} is the minimum average SED for log-type l allowed by customer m;

SED^{max}_{ml} is the maximum average SED for log-type l allowed by customer m;

q^{SEDmin}_{ml} is the value less than the maximum required SED for customer m and log-type l;

q^{SEDmax}_{ml} is the value more than the maximum required SED for customer m and log-type l;

Ξ is the set of logs with SED requirements;

g_{mkt} is the total volume in log-group k for customer m in period t;

G_k is the set of log-types in log-group k;

F^{max}_{ml} is the maximum fraction for log-type l for customer m;

F^{min}_{ml} is the minimum fraction for log-type l for customer m;

q^{Fmax}_{mlt} is the fraction of log-type l exceeding the maximum fraction for customer m in period t;

q^{Fmin}_{mlt} is the fraction of log-type l less than the minimum fraction for customer m in period t;

c_{ci} is the cost of crew c following schedule i;

$c_{P,ml}$ is the revenue from log-type l delivered to customer m;

$c_{T,hml}$ is the transportation cost to move log l from harvest unit h to customer m;

$c_{R,h}$ is the value of the residual area of harvest unit h;

$c_{IR,hl}$ is the value of the residual log-stocks of log-type l left in harvest unit h at the end of the time horizon;

C_{mlt}^{Dmin} is the cost associated with shortfall volume of log-type l in period t for customer m;

C_{mlt}^{Dmax} is the cost associated with excess volume of log-type l in period t for customer m ;

C_{ml}^{SEDmin} is the cost of violating the SED minimum for customer m and log-type l;

C_{ml}^{SEDmax} is the cost of violating the SED maximum for customer m and log-type l;

C_{mlt}^{Fmin} is the cost of the penalty variables on the minimum log fraction constraints;

C_{mlt}^{Fmax} is the cost of the penalty variables on the maximum log fraction constraints.

Chapter 7

Additional Formulation Features

A savage place! as holy and enchanted
As e'er beneath a waning moon was haunted
By woman wailing for her demon-lover!
(Coleridge 1798)

In this chapter, two additional features of the formulation are discussed in detail. They are the relaxation of the integer requirements of a feasible solution and the restrictions on generated crew schedules. Because both these formulation features require some description of the solution process, they do not easily fit with the model formulation discussed in the Chapter 6.

Section 7.1 describes the modification of the integer requirements of a feasible solution. In other integer formulations in the literature, the crew schedule variables (x_{cht}) are binary integer ($x_{cht} \in \{0, 1\}$). In this thesis, this requirement is relaxed to allow crews to move between harvest units *within* a period. The techniques of constraint branching (Section 5.2.3.3) are used to describe a set of mutually compatible crew schedules (J_c) for each crew. This set can be interpreted to allow the crew to move at any time within the planning horizon and not only at the period boundaries.

In Section 7.2, the restrictions on the structure of valid crew schedule variables are explained. These restrictions were discussed briefly in Section 6.2.2 and lead to a more efficient solution process. An example of this type of structure is, a crew schedule must not indicate that a crew will remain in a harvest unit longer than it would need to completely harvest the unit. Imposing this structure tightens the bound on the integer solution given by the optimal objective value of the Relaxed Linear Program (RLP) .

7.1 Integer relaxation

In an OHS solution, each crew must be *unambiguously* located throughout the planning horizon. In this context, unambiguous means that a crew is located in a single harvest unit at every single instant. If this requirement is not enforced, a crew could be spread between several harvest units throughout the forest.

A simple LP formulation of the OHS problem often produces solutions that ambiguously locate crews. To avoid this, the problem is posed as a type of integer programming problem. In the MILP formulations used previously in the literature, the B&B process forces each crew into a *single* harvest unit in *each period* within the planning horizon. Therefore, crews are not able to move between harvest units *within* periods. Some types of crews (for example, hauler crews that only move in weekends, see Section 3.4.2), may be restricted in this manner if a period is a single week. However, if crews are free to move at any time, or periods are longer then a week, this restriction is unnecessary.

In a Model I formulation (see Section 6.2), a crew allocation (see Section 6.1) is determined by a single binary integer variable (x_{cht}). When this variable is at value one, the crew is located in a single harvest unit for that period. Similarly, in a typical Model II formulation, an integer feasible solution will only contain a single non-zero crew schedule (x_{ci}) for each crew. As a crew schedule is a combination of crew allocations, both typical Model I and II formulations suffer from the problems discussed above.

7.1.1 Types of OHS solution

From a forestry perspective, the crew locations of an OHS solution can be interpreted in three ways. Each of these interpretations is in turn related to the structure of the solution from an optimisation viewpoint.

It should be noted that the terms below do not refer to the type of model formulation. The terms relate to solution sets of the OHS model.

In the definitions below a Model II formulation is assumed.

- *Ambiguous location solutions*: Crews' locations are not restricted in any way, a single crew can harvest in many harvest units simultaneously. This is the general classification of all feasible OHS solutions. This solution set is equivalent to the feasible region of the Relaxed Linear Program (RLP).

- *Restricted movement solutions*: Crews are unambiguously located in a single harvest unit in each period. They are only allowed to move between harvest units *between* periods. This is the most restrictive of the three solution sets and a strict subset of the others. This solution set is equivalent to the set of strictly integer solutions ($x_{ci} \in \{0, 1\}$).
- *Unrestricted movement solutions*: Crews are unambiguously located in a single harvest unit at any time within the time horizon. However, the crews may also move between harvest units *within* a period. This type of solution lies in between ambiguous location and restricted movement solutions. This set is a subset of the ambiguous location solution set and ensures the location of the crews is determined at every instant. A relaxed integer solution (see Section 7.1.2) will give solutions within this set.

In summary,

$$\text{Ambiguous location solutions} \supset \text{Unrestricted movement solutions} \supset \text{Restricted movement solutions}.$$

A Model II formulation is necessary to generate the relaxed integer solutions required to find unrestricted movement solutions.

7.1.1.1 Comparison of solution types

Because a crew cannot work in more than one harvest unit at a time, strictly ambiguous location solutions cannot be used in practice. Therefore, the only operationally feasible solutions are the unrestricted and restricted movement solutions.

The harvest unit area constraints (Section 6.4.2.1) in the formulation require that when a crew has harvested all the trees in a harvest unit it must move or stop harvesting. In a restricted movement solution, a crew can only move between periods. If a crew finishes a harvest unit within a period, the crew is idle until the end of the period. Alternatively, the crew can move at the end of the previous period and create an unfinished harvest unit which will never be completed. An unrestricted movement solution does not suffer from these problems as crews will simply move when they finish the harvest unit.

The restricted movement solution set is also the most restrictive of the three. As restricted movement solutions are more constrained, the objective value of the optimal restricted movement solution cannot be better than the optimal objective of the other two solution sets.

7.1.1.2 Construction of unrestricted solutions

An unrestricted movement solution allows crews to move between harvest units *within* a period. To move within a period the crew must be located in two harvest units in the same period (i.e. in harvest unit **A** in the beginning of the period and in **B** at the end). The unrestricted movement solutions in this thesis do not allow a crew to move more than once per period. If it is desirable for a crew to move more often the period length should be reduced.

In a Model I formulation (where a variable represents a crew allocation), it is difficult to model a solution that allows the crews to move, while retaining the ability to locate crews unambiguously. The difficultly arises because of the inability to use historical crew location information in this kind of formulation, without cumbersome linking constraints.

However, an unrestricted movement solution can be obtained by two separate methods in a Model II formulation.

- Allow the column generation algorithm to generate schedules where movements can occur mid-period.
- Interpret linear combinations of crew schedules to indicate that a crew moves between harvest units within a period.

The first option will accurately model movement costs and penalties. Each of the columns generated in this option will precisely indicate when, within the period, the crew will move. However, if a crew moves momentarily earlier or later a new crew schedule is necessary. This small change in movement timing may be caused by changing the cutting strategy used or by other subtle changes. If a column generation algorithm is used it would need to produce a range of very similar crew schedules. As the crew schedules vary continuously, a-priori generation all the columns, even for small problems, will be impossible.

The second method, which is the preferred method in this thesis, is discussed in Section 7.1.2.

7.1.2 Relaxed integer solutions

In an unrestricted movement solution, more than one crew schedule for each crew is allowed to be non-zero in a solution. The combination of these active schedules is interpreted to define a mid-period crew movement. A solution that contains several active crew schedules for each crew, while unambiguously locating them, is known as a *relaxed integer solution*. In a relaxed integer solution, the crew schedule variables are no longer truly integer. This approach reduces the number of variables necessary, while introducing flexibility into the integer decisions that allow schedules to be found that meet the harvest unit area constraints.

7.1.2.1 Generation of relaxed integer solutions

Several crew schedules can combine to form a relaxed integer solution. A set of rules governs which combinations can be interpreted as an unambiguous placement of the crew. Combinations that are allowed in a relaxed integer solution are called *Complementary columns*. Two crew schedules are complementary if they both follow the same sequence of crew movements. In addition, the crew movements in the individual crew schedules can only differ from the sequence by a single period.

For example, in a five period problem crew Skyline harvests two units **A** and **B**. If Skyline finished unit **A** 0.6 of the way through period 3, then the crew schedules needed are shown in Table 7.1.

Table 7.1: Two crew schedules with an unambiguous interpretation.

Crew Skyline	x=0.4	y=0.6
Period 1	A	A
Period 2	A	A
Period 3	B	A
Period 4	B	B
Period 5	B	B

The interpretation of this combination of these variables states that the crew moves from harvest unit **A** to harvest unit **B** 0.6 of the way through period 3.

The crew schedule compatibility requirement is stated precisely below.

Requirement 1 (Crew schedule compatibility) *If a crew schedule indicates that a crew is in harvest unit s in period p, another crew schedule can co-exist in a relaxed integer solution if any of these three conditions are met:*

1. *the second crew schedule indicates harvest unit s in period p;*
2. *the second crew schedule indicates harvest unit t in period p, if the first variable indicates harvest unit t in period $p+1$ (first schedule lags);*
3. *the second crew schedule indicates harvest unit r in period p, if the first variable indicates harvest unit r in period $p-1$ (second schedule lags).*

In the example in Table 7.2, from Requirement 1 crew schedule **y** is compatible with both **x** and **z**. However, crew schedules **x** and **z** are not compatible with each other, because movement penalties are not incurred in the same period.

Table 7.2: Three crew schedules with two different interpretations.

Crew Skyline		x	y	z
	Period 1	A	A	A
	Period 2	A	A	A
	Period 3	B	A	A
	Period 4	B	B	A
	Period 5	B	B	B

In effect, the pairs **(x,y)** or **(y,z)** define a specific sequence of harvest units. The sequences given by both these pairs are not compatible with each other as **(x,y)** indicates the crew moves in period 3 while **(y,z)** indicates the movement is in period 4. To interpret the solution given by the compatible columns correctly, the *crew movement sequence* must be found. This sequence is similar to a crew schedule but specifically defines which period the crew moves. For example, the crew movement sequences for the pairs of variables are shown in Table 7.3.

The movement sequence determines which harvest units the crew is in at the beginning and the end of each period. The fraction Γ_{cht} of the period t that crew c remains in harvest unit h will be the sum of the values of the crew schedules x_{ci} that allocate the crew c to harvest unit h in period t. This relationship is shown in Equation (7.1).

Table 7.3: Movement sequences for pairs of crew schedules

Crew Skyline		Pair x, y	Pair y, z
	Period 1	A	A
	Period 2	A	A
	Period 3	A→B	A
	Period 4	B	A→B
	Period 5	B	B

$$\sum_i x_{ci} H_{ht}^i = \Gamma_{cht} \ \ldots \quad \forall c, h, t \tag{7.1}$$

where:

- x_{ci} is the allocation of crew c to schedule i;
- H_{ht}^i $= 1$ when schedule i contains harvest unit h in period t and zero otherwise;
- i indexes possible schedules;
- c indexes the crews;
- h indexes the harvest units;
- t indexes periods.

It is evident that a single crew schedule may be interpreted in two ways. For example, schedule **y** represents the crew moving at the end of period 3 in movement sequence **(x,y)**. In movement sequence **(y,z)**, schedule **y** represents the crew moving at the beginning of period 4. In the current formulation the crew schedule construction is identical if the crew moves at the beginning of period t or at the end of $t-1$. Therefore, the exact same variable can occur in two different movement sequences.

Not all combinations of crew schedules can be interpreted as an unambiguous location for the crews. For instance, the group of variables shown in Table 7.4 is ambiguous. Though the crew is limited to only two harvest units **A** and **B**, in each period the variables **x** and **y** cannot be interpreted to give an unambiguous location to the crew.

The relaxed integer solution is constructed during the B&B process (see Sections 8.3.1.1 and 8.3.2 for details). In the B&B, the crew movement sequences for all crews are built sequentially period by period. Each branch in the B&B process removes crew schedules that do

Table 7.4: Two crew schedules without an unambiguous interpretation.

Crew Skyline		x	y
	Period 1	A	B
	Period 2	A	B
	Period 3	A	B
	Period 4	A	B
	Period 5	A	B
	Period 6	A	B

not comply with the current incremental crew movement sequence. An unrestricted movement solution is found when a complete crew movement sequence is constructed for all crews.

7.2 Valid crew schedules

In common with other SPP formulations of rostering problems, a major benefit of a Model II formulation is that the construction of the columns (in this formulation columns represent crew schedules) can reflect implicit constraints that are not found in the LP formulation. These implicit constraints guarantee that the only columns available to solve the problem conform to the required structure. A column that obeys all implicit constraints is *valid* while those that don't are *invalid*.

In a Model II formulation, the most useful information available to the column generation algorithm is the history of the crew's location. This information is useful in several ways.

- Only integer feasible crew schedules are generated, thus the RLP solution becomes closer to the best integer solution.
- Crew schedules can be used to force crews to finish harvest units before they move.
- Crew schedules must meet the requirements of any active constraint branches.

7.2.1 Integer feasible columns

When comparing the RLP solutions to the eventual integer feasible solution, it is common to get situations similar to the one illustrated in Table 7.5.

In this example, each harvest unit is completely harvested in both the RLP and the integer

Table 7.5: Contrast of RLP solution variables to eventual integer solution

Crew Skyline		RLP variables $x=\frac{1}{3}$	$y=\frac{1}{3}$	$z=\frac{1}{3}$	Integer solution
	Initial	A	A	A	A
	Period 1	A	B	C	A
	Period 2	A	B	C	A
	Period 3	A	B	C	B
	Period 4	A	B	C	B
	Period 5	A	B	C	C
	Period 6	A	B	C	C

solutions. There is a major difference between the structure of the crew schedules in the RLP solution and the eventual integer solution. In the integer solution, the crew is forced to move twice. In the RLP solution the number of crew movements is $\frac{1}{3} \times 0 + \frac{1}{3} \times 1 + \frac{1}{3} \times 1 = \frac{2}{3}$ of the single movement cost. Because the first crew schedule has no movements, and the second and third crew schedules only move in the initial period. The RLP solution economises on crew movement costs in a way that is not possible in the integer solution.

The reason the RLP solution is so different from the integer solution is that a crew schedule can allow a crew to remain in a single harvest unit for the entire planning horizon. In practice, a crew would finish harvesting this harvest unit before the end of the planning horizon, as shown in Table 7.5. Unstructured crew schedules give a RLP solution significantly different to the integer feasible solution, because the harvest unit area constraints combined with the integer requirements can change the solution significantly during the B&B process.

B&B will eventually find an integer solution to the problem. However, as the bound gap will be large the B&B is likely to be inefficient. The lower bound on the solution value determined by the optimal objective value of the RLP solution with unrestricted crew schedules is not a good indication of the best possible integer solution.

To resolve this issue and obtain a tighter bound from the RLP solutions an implicit constraint was added to the crew schedule structure. The constraint ensures that any crew schedule generated must not violate the harvest area constraints when the crew schedule is at value one.

Requirement 2 (Integer feasible columns) *If crew c will complete harvest unit h in t periods,* *the number of periods h appears in a crew schedule for c may not exceed $t + 1$.*

For example, if a crew would complete a harvest unit in 3.2 periods no crew schedules will be generated with the crew remaining in the harvest unit for more than 4 periods. The addition of the extra period allows relaxed integer solutions to be generated.

The effect of this constraint is to force the structure of the RLP solution to be similar to an integer solution to the problem. The statement above will force crew schedules in the RLP to have a similar structure to the crew schedules in the integer solution and thus give a better bound.

7.2.2 Forcing harvest unit completion

As mentioned in Section 6.4.2.1 in some cases crews may be required to finish a harvest unit once they have entered. This constraint can be enforced in the B&B. However, as in Section 7.2.1 it is desirable that the RLP solutions approximate the eventual integer solutions to the problem. In this case, a similar restriction to Requirement 2 can be enforced to guarantee that crews remain in the harvest units for the minimum time needed to completely harvest them.

Requirement 3 (Force HU completion) *Harvest unit h must be harvested in a single operation. Crew c will take t periods to complete h. The number of periods that h appears in any crew schedule for c may be either 0 or greater or equal to $t - 1$.*

This requirement can only be enforced on harvest units where only a single crew can harvest at any one time. If two crews or more are in a harvest unit simultaneously, the harvest unit will be completed sooner than can be predicted for a single crew. Thus, no variables can be generated with this requirement that allow multiple crews to harvest together.

7.2.3 Enforcing constraint branches

Because the column generation algorithm is called within the B&B tree, all newly generated crew schedules need to comply with constraint branches that have been implemented.

Requirement 4 (Compliance with constraint branches) *All crew schedules generated during the Branch and Bound (*B&B*) process must comply with all constraint branches in effect for the current node.*

Chapter 8

Solution Strategy

And from this chasm, with ceaseless turmoil seething,
As if this earth in fast thick pants were breathing,
A mighty fountain momently was forced:
(Coleridge 1798)

This chapter describes the implementation of the techniques discussed in Chapter 5 that are used to solve the Operational Harvest Scheduling (OHS) problem. These techniques have been implemented in an Operational Harvest Scheduling Algorithm (OHSA) that was used to generate the results presented in this thesis.

The specific techniques discussed in this chapter are:

- Relaxed Linear Program (RLP) solution techniques;
- column generation;
- Branch and Bound (B&B) process;
- integer allocation heuristics.

This chapter also includes a description of the *ZIP* programming environment (Ryan 1980).

8.1 RLP solution strategy

The first stage in the OHSA solves the RLP of the formulation given in Chapter 6. The LP methods used in this thesis are provided by ZIP4.0 (Ryan 1980) that also provides the framework for the Branch and Bound (B&B) algorithm.

8.1.1 ZIP solver

ZIP is the optimisation environment used in this thesis. ZIP provides the backbone of the optimisation algorithm providing an implementation of the simplex algorithm and a branch and bound framework. ZIP allows the user to customise various steps in the solution process. It is written in the *FORTRAN* programming language.

ZIP is used in preference to commercial packages for example, CPLEX (described in CPLEX (1994)), because of the flexibility provided by the ZIP environment. ZIP is flexible because the core routines of ZIP only use the current basis and do not include any problem specific information. This flexibility allows easy implementation of:

- constraint branching (see Section 8.3);
- column generation (see Section 8.2) in the B&B.

The structure of ZIP is divided into three parts as shown in Figure 8.1:

- core routines that remain unchanged between different problems;
- user routines that interface with the core and provide the information needed to solve the problem;
- problem specific routines and data structures that define the optimisation model.

The core routines were unchanged in this thesis.

- *PRIMAL*: applies the simplex algorithm to the problem.
- *BANDB*: applies B&B to the RLP solution.

The user routines are problem specific functions that interface ZIP with the problem. These functions were extensively modified in this thesis.

- *PEVAR*: chooses the entering variable in each iteration of the simplex algorithm.
- *UNPACK*: constructs the matrix column that represents the entering variable.
- *ALLOC*: implements a heuristic method to find an integer solution from a nodal solution in the B&B tree.

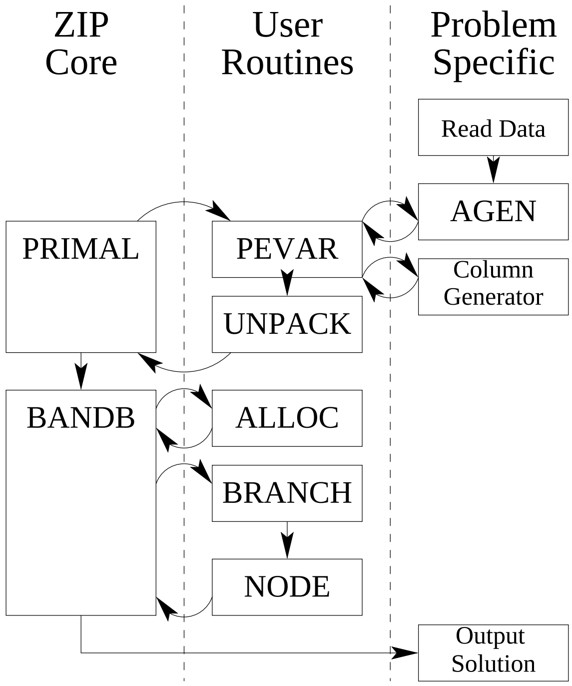


Figure 8.1: The ZIP solution environment

- *BRANCH*: identifies and implements the constraint branches in the B&B tree.
- *NODE*: chooses the active node to be examined in B&B.

The problem specific routines read and display the problem parameters, generate the necessary data structures and interface with the user routines.

- Read Data: reads and pre-processes the OHS parameter information;
- *AGEN*: generates the initial matrix;
- Column Generator: generates new columns with negative reduced costs;
- Output Solution: outputs the solution information.

Within the ZIP solution framework the OHSA :

- provides pre-solve functions;
- controls the calculation of entering variables;
- generates new columns;
- guides the B&B process.

Details of these functions are discussed in the remainder of this chapter.

8.1.2 Problem pre-solve

Specific functions within the OHSA pre-solve process control:

- the scaling of columns and rows of the constraint matrix;
- generation of crew - harvest unit compatibility based on harvest unit size;
- generation of an initial basis.

All of these functions require data for a specific instance of the OHS problem.

8.1.2.1 Scaling

Scaling alters the numerical value of constraint matrix coefficients (see Section 5.1.1) to improve the numerical stability and performance of the LP solution methods. As ZIP was designed primarily to solve scheduling problems that contain matrix coefficients of zero or one, there is no internal scaling mechanism. The formulation of the OHS however, contains a large number of constraints that contain real number coefficients, so scaling of the problem is necessary. Both rows and columns in the constraint matrix are scaled.

Traditional scaling techniques calculate scale factors based on the components of the constraint matrix. However, in the OHSA, the column generation algorithm adds new variables to the problem as the solution algorithm progresses. Therefore, the scaling algorithm cannot examine all the matrix coefficients a-priori and scale factors must be based on parameter information available before column generation. These parameters are used to calculate the likely magnitude of matrix coefficients. If the scale factors are computed in this manner, they need not be recomputed every time a new column is added to the matrix.

Many of the scaling functions in the OHSA occur before the initial matrix is generated for the simplex algorithm.

With scaling removed from the problem, none of the case studies in Chapter 9 would solve due to numerical instability.

8.1.2.2 Crew harvest unit compatibility modification

A relaxed integer solution (Section 7.1) to the OHS problem requires that crews remain in a harvest unit for a minimum of one period once they enter and begin harvesting. If an integer solution is required for a period, a crew's productivity in the period (Section 6.7.3) cannot exceed the initial area of any compatible harvest unit. This restriction will be imposed during the B&B process, if not otherwise modelled.

To ensure the RLP solution is a strong lower bound for the B&B algorithm, the crew harvest unit compatibility data (Section 6.4.2.2) are altered. The changes prevent crews entering harvest units that do not have sufficient initial area for the crew and the period length. The compatibility data are changed before the initial basis is constructed.

The period length is typically a single week in the first few periods of the planning horizon. As the period length is short not many crews will be excluded and the compatibility requirement should not significantly alter the OHS solution. Even without the requirement, if a crew did enter a harvest unit only to leave less then a week later the movement costs for that crew will be very high compared to its productivity. Because of the increase in cost, this behaviour will be unlikely to occur in an optimal OHS solution and less productive crews will be allocated to the smaller harvest units.

The compatibility modification does not apply to the initial harvest units allocated to crews. A relaxed integer solution does allow initial units to be harvested for less than a single period. This behaviour allows a harvest unit to be completely clear-felled when only a small residual area is left at the beginning of the time horizon.

8.1.3 Initial basis selection

To create an initial non-singular basis matrix in AGEN slack and surplus[1] variables from the problem are selected. Together with artificial variables added where necessary, a non-singular matrix with columns of the positive or negative identity ($\boldsymbol{B} = [\pm I_r \ldots]$) is formed. Phase-one simplex removes the artificial variables from the basis if possible, to give a feasible solution that phase-two continues to solve (see Section 5.2.1.1).

The OHSA only 'cold starts' (initiates the simplex method without a previous feasible basis) once so, the phase-one, phase-two method is adequate. However, some small improvement in solution time may result if more sophisticated methods where used to find an initial basis.

[1]Surplus variables are only used where the RHS vector is zero.

8.1.4 Entering variable pricing

The entering variable is found by PEVAR in each iteration of the simplex method (Section 5.2.1). Therefore, efficient methods that find an entering variable quickly will decrease the overall solution time of the OHS problem.

There are two main types of variables in the OHS problem. The crew schedule variables (see Section 6.7.5) are generated by the column generator. All other variables are generated a-priori in AGEN. These variables include the:

- volume allocation variables;
- downgrade linking variables;
- inventory linking variables;
- and slack and surplus variables.

Within the OHSA all variables are stored in a compressed format, the coefficients of the column that represents the variable are not calculated until the variable enters the basis and UNPACK is called.

The search for the entering variable has three stages, the algorithm only proceeds to a successive stage if no entering variables are found in the previous stages.

1. Crew schedules that are already included in the LP matrix are partially priced by crew partition. The various slack, surplus, or linking variables generated a-priori are priced as a separate partition.
2. The column generation algorithm generates new crew schedules based on new combinations of crew allocations.
3. The current solution is declared optimal. New yield predictions may be generated externally and the problem re-solved. When the problem is re-solved the algorithm returns to stages 1 and 2. New crew schedules may be generated from a combination of existing and new yield predictions.

8.1.4.1 Reduced cost components ζ_{hst}

To reduce the computational effort required to find the entering variable, the reduced cost calculation is made more efficient by the use of reduced cost components (ζ_{hst}). Though the technique described does speed the computation of the reduced cost for crew schedules that have already been added to the LP matrix, the real savings from using ζ_{hst} are found in the column generation algorithm (see Section 8.2).

The reduced cost of a crew schedule variable (x_{ci}), with the associated column $\boldsymbol{a}_{ci}$, is defined from Equation (5.5),

$$rc_{ci} = c_{ci} + \boldsymbol{\pi}^T \boldsymbol{a}_{ci} \ldots \forall c, i.$$

If we partition the $\boldsymbol{\pi}^T$ vector in the same way as we partition the $\boldsymbol{a}_{ci}$ vector in Equation (6.23) we get

$$\boldsymbol{\pi}^T = \left[(\boldsymbol{\pi}^C)^T, (\boldsymbol{\pi}^H)^T, (\boldsymbol{\pi}^A)^T, (\boldsymbol{\pi}^P)^T, (\boldsymbol{\pi}^N)^T\right].$$

Therefore,

$$rc_{ci} = c_{ci} - (\boldsymbol{\pi}^C)^T \boldsymbol{a}_{ci}^C - (\boldsymbol{\pi}^H)^T \boldsymbol{a}_{ci}^H - (\boldsymbol{\pi}^A)^T \boldsymbol{a}_{ci}^A - (\boldsymbol{\pi}^P)^T \boldsymbol{a}_{ci}^P \quad \ldots \forall c, i. \tag{8.1}$$

The value of rc_{ci} is calculated every time a crew schedule (x_{ci}) is priced in PEVAR. If a large number of crew schedules share crew allocations, many calculations will be repeated if the full form of Equation (8.1) was used to calculate rc_{ci}. The repeated calculations occur because crew schedules that share crew allocations will have many identical coefficients in their matrix columns.

We can take the components of the $\boldsymbol{\pi}^T$ vector and transform them into subscripted variables. The subscripts are based on the significance of the appropriate rows of the constraint matrix.

$$\begin{aligned} (\boldsymbol{\pi}^C)^T &= [\pi_1^C \ldots \pi_{|C|}^C] \\ (\boldsymbol{\pi}^H)^T &= [\pi_{1,1}^H \ldots \pi_{1,|T|}^H, \pi_{2,1}^H \ldots \pi_{|H|,|T|}^H] \\ (\boldsymbol{\pi}^A)^T &= [\pi_{1,1}^A \ldots \pi_{1,|T|}^A, \pi_{2,1}^A \ldots \pi_{|H|,|T|}^A] \\ (\boldsymbol{\pi}^P)^T &= [\pi_{1,1,1}^P \ldots \pi_{1,1,|T|}^P, \pi_{1,2,1}^P \ldots \pi_{|H|,|L|,|T|}^P] \end{aligned} \tag{8.2}$$

We can then find rc_{ci} by the appropriate summation and multiplication of the column coefficients H_{ht}^i, A_{ht}^{ci} and P_{hlt}^{ci} (see Section 6.7) with the components of $\boldsymbol{\pi}^T$.

$$rc_{ci} = c_{ci} - \pi_c^C - \sum_{\substack{t \\ h=H(i,t)}} \left(\pi_{ht}^H + \pi_{ht}^A A_{ht}^{ci} + \sum_l \pi_{hlt}^P P_{hlt}^{ci} \right) \ldots \forall c, i.$$

If we expand the column coefficients by using Equations (6.19)–(6.21)

$$\begin{aligned} rc_{ci} = & \sum_{\substack{t \\ h=H(i,t) \\ h'=H(i,t-1)}} P_{ch} M_{h'h}^P C_{ch} + \sum_{\substack{t \\ h=H(i,t) \\ h'=H(i,t-1)}} M_{h'h}^{\$} - \pi_c^C \\ & - \sum_{\substack{t \\ h=H(i,t) \\ h'=H(i,t-1) \\ s=S(i,t)}} \left(\pi_{ht}^H + \pi_{ht}^A H_{ht}^i P_{ch} M_{h'h}^P A_{Y,hs} + \sum_l \pi_{hlt}^P H_{ht}^i P_{ch} M_{h'h}^P P_{F,hsl} \right) \\ & \ldots \forall c, i. \end{aligned}$$

Rearranging,

$$\begin{aligned} rc_{ci} = & -\pi_c^C + \\ & \sum_{\substack{t \\ h=H(i,t) \\ h'=H(i,t-1) \\ s=S(i,t)}} \left[M_{h'h}^{\$} - \pi_{ht}^H + P_{ch} M_{h'h}^P \left(C_{ch} - \left[\pi_{ht}^A A_{Y,hs} + \sum_l \pi_{hlt}^P P_{F,hsl} \right] \right) \right] \\ & \ldots \forall c, i. \end{aligned}$$

The reduced cost can then be calculated by

$$rc_{ci} = -\pi_c^C + \sum_{\substack{t \\ h=H(i,t) \\ h'=H(i,t-1) \\ s=S(i,t)}} \left[M_{h'h}^{\$} - \pi_{ht}^H + P_{ch} M_{h'h}^P \left(C_{ch} - [\zeta_{hst}] \right) \right] \ldots \forall c, i \qquad (8.3)$$

where:

ζ_{hst} is the contribution to the reduced cost given by placing a crew, using cutting strategy s, into harvest unit h in period t.

$$\zeta_{hst} = \pi_{ht}^{A} A_{Y,hs} + \sum_{l} \pi_{hlt}^{P} P_{F,hsl} \ldots \forall h, s, t \tag{8.4}$$

Equation (8.4) shows the contributions (ζ_{hst}) are the dual variables for the area constraint times the area harvested ($A_{Y,hs}$), plus the sum of the dual variables for the production constraint times the log production $P_{F,hsl}$ for all logs.

Note, that ζ_{hst} is independent of both the crew and previous harvest unit. The ζ_{hst} values must be multiplied by the crew productivity (P_{ch}) and movement penalties ($M^{P}_{h'h}$) to find the actual contribution to these constraints.

Equation (8.3) calculates the reduced cost by multiplying ζ_{hst} by P_{ch}, then adding crew costs (P_{ch} C_{ch}), multiplying by the movement penalty ($M^{P}_{h'h}$) then adding movement costs ($M^{\$}_{h'h}$). The contribution (-$\pi^{H}_{ht}$) to the harvest unit capacity is added. The value is summed over all the crew allocations in the time horizon, and the contribution (-π^{C}_{c}) to the crew allocation constraint is included.

The values of ζ_{hst} are calculated once before entering variables are priced. These values are used to price existing variables as well as within the column generation strategy. When ζ_{hst} is pre-calculated there is a saving of $1 + 2|L|$ floating point operations for each harvest unit, strategy, period allocation that is shared between two crew schedules.

8.2 Column generation

Chapter 5.2.2 discusses how column generation allows a set-partitioning problem with many variables to be solved in reasonable time. Column generation achieves this by efficiently pricing variables that have not previously been included in the problem.

8.2.1 Problem description

The column generation sub-problem in the OHSA finds crew schedules that have a negative reduced cost. It is formulated as the shortest path problem shown diagrammatically in Figure 8.2. Each node represents a possible harvest unit allocation for a crew. The arcs represent the transition between periods where a crew moves between harvest units or remains in place. The cost of each arc is the incremental contribution of the transition to the reduced cost of the crew schedule. This contribution is the value generated from the new allocation plus any penalties and costs of movement. Most arc costs will be negative.

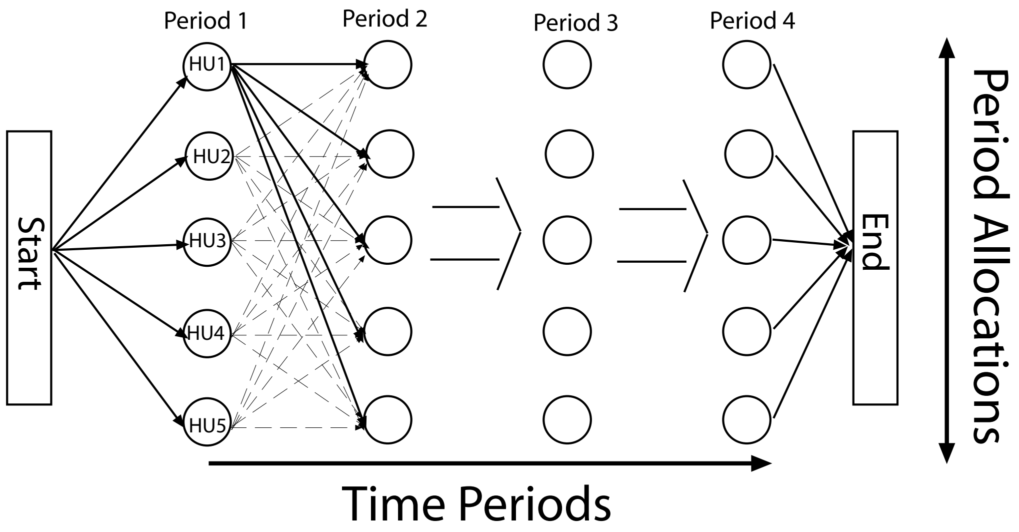


Figure 8.2: The shortest path formulation of the column generation

In the shortest path formulation, arcs join nodes in successive periods. The existence of an arc implies that it is feasible to move the crew from harvest unit h' to harvest unit h in the period (Section 6.4.2.2). Removal of arcs that do not comply with active branches (Section 8.3) ensures that the column generation process does not generate variables that been banned by constraint branches. Other *implicit* (see Section 7.2) constraints on crew schedule structure are applied in a similar way within the column generation rather than within the LP formulation.

8.2.2 Dynamic programming algorithm

In the shortest path formulation, the incremental costs on the arcs in period t are independent of the nodes that have been visited earlier then $(t-2)$. The *Principle of Optimality* (Section 5.2.2.1) is valid and a DP recursion can be used to find the entering variable.

The DP recursion uses periods as the stages and the harvest unit, cutting strategy allocation as the state. Note, the selection of cutting strategy for a harvest unit is not affected by previous decisions, so the cutting strategy can be determined for each harvest unit independently of the DP.

The DP algorithm for the column generator in the OHSA is given in Algorithm 8.1. The initial DP formulation does not include the integer feasible crew schedule considerations (Section 7.2.1) that will make the problem a resource constrained shortest path problem. These issues are discussed in Section 8.2.4.

The values of ζ_{hst} are used to calculate the arc costs $l_{ch'ht}$. These values can be pre-calculated and are the incremental contribution for each allocation (see Section 8.1.4.1). The cost of each arc is therefore

$$l_{ch'ht} = M^{\$}_{h'h} - \pi^H_{ht} + (C_{ch} - \zeta_{hst})P_{ch}M^P_{h'h} \tag{8.5}$$

where:

s	is the optimal cutting strategy for harvest unit h in period t;
$l_{ch'ht}$	is the arc cost for crew c to move from h' to h and using the optimal strategy in period t.

Note, that Equation (8.5) is simply the summand of Equation (8.3).

Valid harvest units are determined by the data on legitimate harvest units for the crew in each period and by any applied branches.

Algorithm 8.1 Column generation DP recursion

Require: Reduced cost components ζ_{hst}
Require: Minimum cost cutting strategy s for all stands and periods s (h,t). { Calculated from $\boldsymbol{\pi}^T$.}
for Each crew, c **do**
  $d_{ht} = -\pi^C_c \ldots \forall h, t$ {distances on shortest path}
  $p_{ht} = 0 \ldots \forall h, t$ {predecessors}
  for Each period, t **do**
    for Each valid harvest unit, h **do**
      Use the minimum cost strategy s =s (h,t)
      Use LIMITSTANDS to reduce the number of candidate predecessors H' (Section 8.2.3).
      Find $(h')^* \in \arg\min_{h' \in H'} \left\{d_{ht} = d_{h'(t-1)} + l_{ch'ht}\right\}$
      $p_{ht} = (h')^*$
      $d_{ht} = d_{(h')^*(t-1)} + l_{c(h')^*ht}$
    end for
  end for
  Create similar crew schedules (Section 8.2.5).
  Add all generated crew schedules with negative reduced cost to the LP matrix.
end for

8.2.3 Reduction of candidate predecessors

As the DP recursion works forward through the time horizon, at each period a unique predecessor harvest unit $((h')^*)$ must be found for each harvest unit. The LIMITSTANDS function reduces the set of eligible predecessor harvest units (H') that are evaluated at each stage of the DP recursion. The place of LIMITSTANDS within the column generation algorithm is shown in Algorithm 8.1.

In a conventional DP iteration, the path from each state in the previous stage to the current state and stage is considered to find the best-cost path. Examining every possible state could be time consuming if the cost calculation is expensive, the number of stages or states is large or the DP is called repetitively. In the OHSA, the column generation is called repeatedly and the number of states is related to the number of harvest units. Therefore, a reduction in the number of predecessors considered at each state and stage in the dynamic program is a useful method of decreasing the column generation time within the problem.

Table 8.1: Comparison of scenarios with and without predecessor reduction

	NZCop_Base	NZCop_NotPR	NZC_FC	NZC_NotPR	AC_Base	AC_NotPR
RLP Objective	605,331	605,331	18,292,376	18,292,398	1,598,726	1,598,726
Objective	583,776	583,776	18,180,609	18,167,752	1,554,879	1,554,879
Objective without penalty ($)	1,475,780	1,475,780	18,681,075	18,672,975	1,906,526	1,906,504
Value of demand violation ($)	446,002	446,002	250,233	252,611	110,277	110,271
Bound gap (%)	3.56	3.56	0.61	0.68	2.74	2.74
Solve time (secs)	8.47	8.56	159.64	189.17	154.2	148.45
Column Generation time (secs)	0.08	0.14	49.27	65.67	2.83	3.67
CG time per variable (msecs)	0.08	0.17	1.58	1.86	0.85	1.11

Table 8.1 summarises the solutions of several scenarios generated by the OHSA the row columns for the data are:

- RLP Objective: The objective value of the RLP solution multiplied by -1 to give the solution profit minus any applied penalties;
- Objective: The objective value of the integer solution multiplied by -1 to give the solution profit minus any applied penalties;

- Objective without penalty: The objective value of the integer solution without subtraction of penalty values. This represents the dollar value of the solution;
- Value of demand violation: The volume of shortfall or excess for each log-type times its price.
- Bound gap: The percentage difference between objective values of the best integer solution and the RLP solution.
- Solve time: The time taken in seconds for the algorithm to finish.
- Column Generation time: The time taken in seconds for the column generation algorithm.
- CG time per variable: The Column Generation Time divided by the number of variables generated in milliseconds.

The three case studies (NZCop, NZC, AC) used in Table 8.1 are explained in detail in Chapter 9. Each case study scenario is solved twice, the solutions found without predecessor reduction are labelled ‘NotPR’. Table 8.1 shows that this method reduces the column generation time moderately but unfortunately does not reduce the overall solution time by very much. The RLP objective values with and without predecessor reduction are equal showing that there is no loss in optimality.

The LIMITSTANDS function reduces the number of predecessors (h') considered at each stage by using known relationships between the predecessors. LIMITSTANDS does not use any forestry specific information to limit the predecessors but relies on the properties of the cost calculation shown in Equation (8.5). LIMITSTANDS does not result in any loss in optimality for the DP recursion.

LIMITSTANDS uses two properties of the state cost calculation

$$d_{ht} = d_{h'(t-1)} + l_{h'h}.$$

These properties are:

- As the cost of the previous state $(d_{h'(t-1)})$ increases the cost of the current state increases. So if $l_{ch'ht}$ is held constant, we can state that if $d_{A(t-1)} < d_{B(t-1)}$ the calculated cost to reach the current state from harvest unit **A** must be less than the cost from **B**;
- The same argument holds for the arc cost $l_{ch'ht}$ so if the costs of the previous states are identical, we can state that if $l_{cAht} < l_{cBht}$ the calculated cost to reach the current state from harvest unit **A** must be less than the cost from **B**.

Distance is a suitable proxy for $l_{ch'ht}$ as the movement costs increase as distance increases, and the movement costs are the only component of $l_{ch'ht}$ that depends on h',

$$D_{Ah} < D_{Bh} \implies l_{cAht} > l_{cBht} \ldots \forall c, h, t, (A, B) \in H.$$

The input data to LIMITSTANDS are:

- the current harvest unit h (state);
- the current period t (stage);
- a ranking of the predecessor harvest units h' ordered by $d_{h'(t-1)}$;
- a ranking of harvest units h' ordered by distance ($D_{h'h}$) into h,

where:

h is the current harvest unit;

h' is a candidate predecessor harvest unit;

$d_{h'(t-1)}$ is the cost to get to the predecessor in the previous period;

$D_{h'h}$ is the distance between the current harvest unit and the predecessor.

LIMITSTANDS will output the reduced set of eligible predecessor harvest units (H'). Note that

$$H' \subset H.$$

The rankings are efficient to calculate in the recursion, as $D_{h'h}$ is constant and the same values of $d_{h'(t-1)}$ will be used in every harvest unit calculation for the current period.

The operation of LIMITSTANDS can be illustrated by considering the three examples given in Figure 8.3, which consists of three possible cross sections from Figure 8.2. In this illustration, the shaded circles represent possible predecessor harvest units (h'). The circles that are not filled represent the current harvest unit (h) that is examined in the current period. The lightest shaded circles represent harvest units that need not be considered when finding $(h')^*$ i.e., $h' \notin H'$.

Consider the simplest scenario where there is no penalty to move between harvest units. This is equivalent to

$$D_{h'h} = 0 \ldots \forall h'h.$$

In this case, only the minimum cost legal predecessor

$$h^* \in: \arg\min_{h' \in H} \left(d_{h'(t-1)}\right),$$

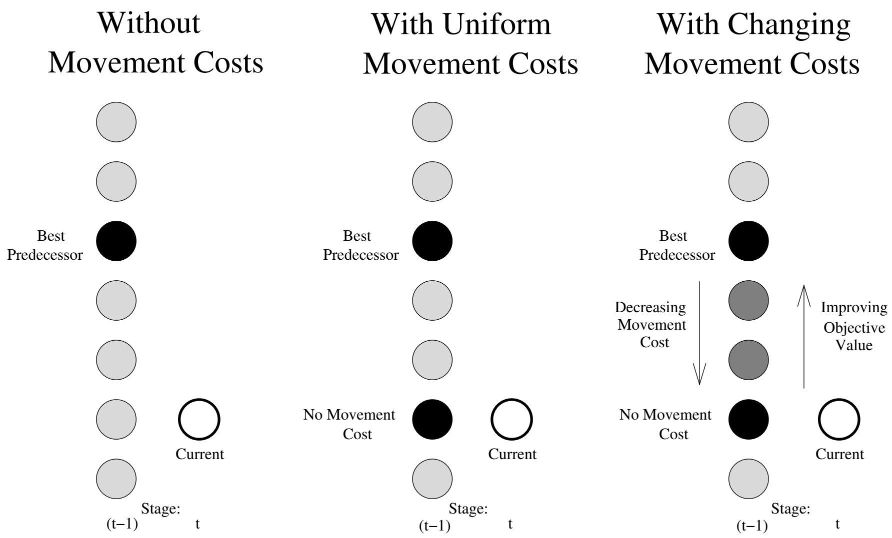


Figure 8.3: Three illustrative cases for the Limit Stands Algorithm

in the previous stage is considered, as all other choices will give a worse d_{ht} for the current harvest unit and period.

In the second scenario, the penalty applied for a movement between harvest units is constant. This is equivalent to

$$D_{h'h} = \begin{cases} 0 \iff h' = h \\ C \qquad \text{otherwise} \end{cases} \quad \ldots \forall h', h.$$

Therefore, only two predecessors are considered. The first is the current harvest unit in the previous period $h' = h$ with no movement penalty. The other predecessor is h^*, which will incur a movement penalty.

The third scenario is the most appropriate to the formulation of the OHS problem detailed in Section 6.7.1. The movement penalty increases with the distance between harvest units. This is equivalent to

$$D_{h'h} = \begin{cases} 0 & \iff h' = h \\ 0 < D_{h'h} < \infty & \text{otherwise} \end{cases} \quad \ldots \forall h', h.$$

Therefore, predecessors (h') with a worse value than the best-cost predecessor ($d_{h'(t-1)} < d_{h^*(t-1)}$) may eventually give a better overall cost (d_{ht}) for the current harvest unit and period. This occurs only if the distance between the harvest units is less than the distance between the current and best cost harvest units. However, predecessors with a greater distance

($D_{h'h} > D_{h^*h}$) than the best-cost predecessor need not be considered, as these predecessors would be dominated by h^*. Similarly, predecessors with a worse cost than the current harvest unit in the previous period $d_{h'(t-1)} > d_{h(t-1)}$ are also disregarded, these would be dominated by h. Therefore, the only stands ($h' \in H'$) that need to be considered are those that are closer than $h^*$and with a better cost then the current stand in the previous period. The set H' is defined below.

$$H' = \{h' \colon (D_{h'h} \leq D_{h^*h} \text{ and } d_{h'(t-1)} \leq d_{h(t-1)})\}$$

where:

H' is the set of predecessor harvest units considered by the column generation.

If restrictions are placed on harvest units that can be legally chosen (either from the branch and bound process, or by the input data to the problem) these must be considered. These legality considerations affect the choice of the minimum cost and minimum penalty predecessors.

Algorithm 8.2 LIMITSTANDS function

Require: h is the current harvest unit.
Require: $C(i)$ The list of valid harvest units (h') ordered by increasing cost in previous period ($d_{h'(t-1)}$).
Require: $M(i)$ The list of valid harvest units (h') ordered by increasing distance ($D_{h'h}$) from harvest unit h.
Find $i_c : C(i_c) = M(1)$
Find $i_m : M(i_m) = C(1) = h^*$
if $\{i_c \leq i_m\}$ **then**
  $H' = C(1 : i_c)$
else $\{i_c > i_m\}$
  $H' = M(1 : i_m)$
end if

8.2.4 Integer feasible columns

In Section 7.2.1, the concept of integer feasible columns was discussed. Integer feasible columns are crew schedules that do not violate the harvest unit area constraints. Therefore, if a crew will finish a harvest unit in three periods an integer feasible column will not allocate the crew to the harvest unit for six periods. The column generation algorithm given in Algorithm 8.1 however, will generate integer infeasible columns, because harvest unit area is not considered. In fact, the column generator will tend to produce integer infeasible columns until the B&B process gradually forces integer feasibility on the problem.

As discussed in Section 7.2.1, while the optimal solution can be found when integer infeasible columns are generated, it is more efficient to only generate integer feasible columns. Adding the implicit constraint detailed in Requirement 2 (see Section 7.2.1) in the column generation will ensure only integer feasible crew schedules are generated. To include this constraint the area of the harvest units must be included in the DP. A new condition must be added to determine the legality of an arc, so a crew can only remain in a harvest unit if there is area left to harvest. Therefore, each state in each stage must have an associated vector of the areas harvested in each harvest unit, to reach the stage.

In addition, two new steps were also added at the end of the period iteration of the DP.

- Transfer the areas harvested from the predecessor state, to the current state.
- Reduce the area of the current harvest unit by the area harvested in the current period.

Table 8.2: Comparison of scenarios with and without using integer feasible columns

	NZCop_Base	NZCop_NotIF	NZC_Base	NZC_NotIF	AC_Base	AC_NotIF
RLP Objective	605,331	611,005	19,175,130	19,204,749	1,598,726	1,598,726
Objective	583,776	583,776	19,156,934	19,103,259	1,554,879	1,554,879
Objective without penalty ($)	1,475,780	1,475,780	19,623,611	19,570,312	1,906,526	1,906,665
Value of demand violation ($)	446,002	446,002	233,338	233,526	110,277	110,308
Bound gap (%)	3.56	4.46	0.09	0.53	2.74	2.74
Solve time (secs)	8.47	8.53	77.95	109.75	154.2	145.53

In Table 8.2, the solutions generated without using integer feasible columns are labelled 'NotIF'. Table 8.2 shows the effects of removing the integer feasibility requirement from the case studies discussed in Chapter 9. The RLP solution value for each case is lower or equal when integer feasible columns are required. The integer objective is however the same or larger for these cases. Therefore, the bound gaps are lower when integer feasibility is required, and the B&B should be more efficient.

Considering the area of the harvest units in the simple manner described above unfortunately, results in a non-optimal DP recursion. To restore the guarantee of optimality the remaining area of each harvest unit will have to be included as a resource in a resource constrained DP.

If the residual area of each harvest unit is considered as a resource, the number of states considered in each DP recursion quickly becomes unmanageable. The number of states in each stage can be described in terms of the lattice path problem (see Appendix C). A full formulation that includes the residual harvest unit area is therefore impractical to use when solving realistic problems. However, the use of near optimal techniques such as merging, may be appropriate.

In some special cases, the simple inclusion of area described above will cause the problem to become infeasible after a crew-harvest unit branch. The infeasibility is caused by the column generation not generating the required feasible crew schedules. As discussion of this situation requires an understanding of crew-harvest unit branches (see Section 8.3.1.1), it is examined in detail in Section 8.3.4. In general, failure occurs when a crew schedule must delay entering a harvest unit so that the harvest unit can be harvested in later periods.

8.2.5 Generation of new columns based on results from the Column Generation

To counter the specific problems mentioned above and discussed in detail in Section 8.3.4, the OHSA generates additional columns based on the columns produced by the column generator, then adds these columns into the LP matrix. We define $N(x_{ci})$ as the set of crew schedules that are similar to x_{ci}. The exact definition of $N(x_{ci})$ in my algorithm is given below.

As the crew schedules in $N(x_{ci})$ are similar to x_{ci}, they are likely to have negative reduced costs as well. In this thesis only complementary columns are generated in $N(x_{ci})$ (see Section 7.1), as these columns are likely to co-exist in the relaxed integer solution to the problem. Therefore, $N(x_{ci})$ is a subset of the complementary columns of x_{ci}.

In the specific example in Section 8.3.4 the failure to generate the required crew schedule leads to infeasibility after a crew-harvest unit branch is imposed. The required crew schedule must delay entering the harvest unit by one period. However, if the required crew schedule has already been added because it is similar to one previously generated, it will already be in the LP matrix and feasibility will be maintained after the branch.

One simple implementation is for $N(x_{ci})$ to contain two crew schedules, one in which all the movements occur a period earlier then the original schedule (x_{ci}) and another where all the movements occur a period later than x_{ci}. The new columns are tested to determine if they comply with imposed branches and other constraints, those that are valid are added to the LP. Other schemes could generate an even wider selection of variables. For instance, $N(x_{ci})$ could include all the complementary columns of x_{ci}. Generating a wide selection of columns will ultimately have the additional advantage of reducing calls to the column generation process.

8.3 Branch and Bound

A general discussion of the Branch and Bound (B&B) algorithm is given by Section 5.2.3. In the OHSA, constraint branching is used (see Section 5.2.3.3). It is also pertinent to note the discussion in Section 6.2.4 that states that constraint branches in a Model II formulation can be interpreted in the same way as variable branches in a Model I formulation.

8.3.1 Definition of branch

In the OHSA there are two types of constraint branch.

- *Crew-harvest unit branch*: allocates a harvest unit to a crew.
- *Crew-strategy branch*: allocates a cutting strategy to a crew and harvest unit.

The allocation of a crew to a cutting strategy without a harvest unit is not considered, because a cutting strategy is meaningless unless applied to a harvest unit. The one-branch of each type will enforce the allocation. The zero-branch will remove any variables with the allocation.

8.3.1.1 Crew-harvest unit branch

A crew-harvest unit one-branch allocates the crew unambiguously to a harvest unit in a particular period. When crew-harvest unit branches are used to find a relaxed integer solution (see Section 7.1) the definition is complicated. For this reason the crew harvest branch is first introduced by considering a simple branch that will find strictly integer solutions.

A crew harvest unit branch creates subsets of crew schedules for a particular crew.

- $J_1^H(cht)$ contains the set of legal crew schedules for a crew after a one-branch.

- $\overline{J_1^H(cht)}$ contains the crew schedules that are removed by a one-branch.
- $J_0^H(cht)$ contains the set of legal crew schedules for a crew after a zero-branch.
- $\overline{J_0^H(cht)}$ contains the crew schedules that are removed by a zero-branch.

If a strictly integer solution is required the one branch ($J_1^H(cht)$) on crew c, harvest unit h, in period t, would simply only allow crew schedules where crew c was harvesting harvest unit h in period t. The zero-branch ($J_0^H(cht)$) allows only crew schedules where crew c was harvesting elsewhere in period t. The implementation of the branches is discussed in detail in Section 8.3.3.

A simplistic crew-harvest unit branch definition for crew c, harvest unit h in period t is shown below.

Simple crew-harvest unit branch:

One) $J_1^H(cht) = \{x_{ci} \colon H(i,t) = h \ldots \forall i\}$

Zero) $J_0^H(cht) = \{x_{ci} \colon H(i,t) \neq h \ldots \forall i\}$

With this simple branch definition, $J_1^H(cht)$ and $\overline{J_0^H(cht)}$ are equal, as are $J_0^H(cht)$ and $\overline{J_1^H(cht)}$.

This simple branch is unsuitable when a relaxed integer solution is required. The OHSA finds a relaxed integer solution by allowing only complementary columns (see Section 7.1) to co-exist in a solution. The crew-harvest unit branch is the mechanism that forces this structure on the solution.

A crew-harvest unit branch can be used to build a movement sequence as defined in Section 7.1.2.1. In fact, as the branches are usually applied sequentially this interpretation is the most logical. The two types of movement sequences a and b allowed by a crew-harvest unit one-branch on harvest unit B in period 3 are shown in Table 8.3 (note that X represents any harvest unit or movement). While c, d and e represent types of movement sequences that are allowed in a crew-harvest unit zero-branch on harvest unit B in period 3 (note !B represents any harvest unit that is not B).

The crew-harvest unit branch definition used in the OHSA is shown below for crew c, harvest unit h in period t.

Crew-harvest unit branch:

Table 8.3: An example of the movement sequences allowed by a branch on B in period 3

period	a	b	c	d	e
1	X	X	X	X	X
2	X	X	!B	B $\rightarrow$!B	X
3	**B**	**X $\rightarrow$ B**	**!B**	**!B**	**!B**
4	X	X	!B	X	!B $\rightarrow$ B
5	X	X	X	X	X
6	X	X	X	X	X
7	X	X	X	X	X
8	X	X	X	X	X

One) $J_1^H(cht) = \{x_{ci} \colon H(i,t) = h \text{ or } (H(i,t) = h' \text{ and } H(i,t+1) = h) \ldots \forall i\}$
$\overline{J_1^H(cht)} = \{x_{ci} \colon H(i,t) \neq h \text{ and } (H(i,t) \neq h' \text{ or } H(i,t+1) \neq h) \ldots \forall i\}$

Zero) $J_0^H(cht) = \{x_{ci} \colon H(i,t) \neq h \text{ or } (H(i,t) = h \text{ and } H(i,t+1) \neq h) \ldots \forall i\}$
$\overline{J_0^H(cht)} = \{x_{ci} \colon H(i,t) = h \text{ and } (H(i,t) \neq h \text{ or } H(i,t+1) = h) \ldots \forall i\}$

where:

$H(i,t)$ is the harvest unit harvested in period t by schedule i;
h' is the harvest unit branched on in period $(t-1)$.

Note, there are some crew schedules that remain in the problem after both a one and a zero-branch. Therefore, $J_1^H(cht) \cap J_0^H(cht)$ is not empty. In fact,

$$J_1^H(cht) \cap J_0^H(cht) = \{x_{ci} \colon (H(i,t) = h \text{ and } H(i,t+1) \neq h) \\ \text{or } (H(i,t) = h' \text{ and } H(i,t+1) = h) \ldots \forall i\}. \quad (8.6)$$

Crew schedules within this intersection can either have $H(i,t) = h$ or $H(i,t) = h'$. When $H(i,t) = h$ the schedule can be interpreted as either a or d types of movement sequence shown in Table 8.3. When $H(i,t) = h'$ the schedule can be interpreted as either b or e and the crew schedules are not removed by either side of the branch. However, subsequent branches will remove crew schedules until only single movement sequence interpretation remains for each crew.

Table 8.4 shows a set of variables for a crew that remain in the problem after a crew-harvest unit branch on harvest unit A in period one (A1) followed by another crew-harvest unit branch on harvest unit A in period two (A2). The table examines the effects of a crew-harvest unit

Table 8.4: Crew schedules in an example crew-harvest unit branch, B3

period	V_1	V_2*	V_3*	V_4*	V_5*	V_6
1	A	A	A	A	A	A
2	A	A	A	A	A	A
3	**A**	**B**	**A**	**B**	**A**	**A**
4	A	B	B	C	B	A
5	A	B	B	C	B	B
6	A	B	B	C	C	B
7	A	B	B	C	C	C
8	A	B	B	C	C	C

branch on the crew in harvest unit B in period three (B3). The crew schedules indicated by stars are in $J_1^H(cB3) = \{V_2, V_3, V_4, V_5\}$ and will remain in the problem while, $\overline{J_1^H(cB3)} = \{V_1, V_6\}$ will be removed. If a zero-branch was applied ($\overline{\text{B3}}$), $J_0^H(cB3) = \{V_1, V_4, V_5, V_6\}$. In this example, $J_1^H(cB3) \cap J_0^H(cB3) = \{V_4, V_5\}$.

8.3.1.2 Crew-strategy branch

If the crew must limit the use of different cutting strategies in a period, crew-strategy branches are employed. A crew-strategy branch allocates a cutting strategy to a crew, harvest unit and period, not just to a crew and a period. There are a number of reasons why the branch is defined in this manner:

- A cutting strategy should always be applied to a harvest unit;
- Multiple crews may work in a harvest unit and not share a cutting strategy;
- A crew may shift harvest units within a period and should not be forced to use the same cutting strategy in both.

The crew-strategy branch can be defined in two ways depending on whether only a single cutting strategy is allowed or multiple complementary strategies are permitted (see Section 6.4.3).

If only a single strategy is allowed. The crew-strategy branch for a single strategy for crew c, harvest unit h, strategy s in period t will create two sets of crew schedules for the crew $J_1^S(chst)$ and$J_0^S(chst)$.

Crew-strategy branch:

One) $J_1^S(chst) = \{x_{ci} \colon S(i,t) = s$ and $H(i,t) = h \ldots \forall i\}$

Zero) $J_0^S(chst) = \{x_{ci} \colon S(i,t) \neq s$ or $H(i,t) \neq h \ldots \forall i\}$

For this branch, like the simple crew-harvest unit branch $J_1^S(chst)$ and $J_0^S(chst)$ are complements.

When multiple compatible strategies are allowed, the crew-strategy branch allocates a set of complementary strategies Q_k, for crew c, harvest unit h, strategy group k in period t.

Complementary Crew-strategy branch:

One) $J_1^{Sc}(chst) = \{x_{ci} \colon (S(i,t) \in Q_k)$ and $H(i,t) = h \ldots \forall i\}$

Zero) $J_0^{Sc}(chst) = \{x_{ci} \colon (S(i,t) \notin Q_k)$ or $H(i,t) = h \ldots \forall i\}$

where:

k	indexes the sets of complementary cutting strategies;
Q_k	is the set of complementary strategies;
$S(i,t)$	is the strategy used in period t by schedule i.

Again, the sets $J_1^{Sc}(chst)$ and $J_0^{Sc}(chst)$ are complements.

8.3.2 Choice of branch

Section 5.2.3 discusses how each unexplored node in the B&B tree must have an appropriate branch and direction. The branch and direction are identified within BRANCH. The motivation for branch choices in the OHSA is to quickly find good integer solutions, and therefore, a depth first search is suitable.

To aid the following discussion some terms need to be defined.

- *Harvest unit integer feasible*: A property of a period that complies with a relaxed integer solution (see Section 7.1).
- *Strategy integer feasible*: A property of a period that satisfies the integer requirements for cutting strategies (see Section 6.4.3).
- *Integer period*: A period that is required to be harvest unit integer feasible and strategy integer feasible. Normally, only the last periods of the OHS are not integer periods.

The choice of branch at each node is outlined in Algorithm 8.3.

Algorithm 8.3 Branch choice decision algorithm

for t =1 to number of integer periods **do**
  if Nodal solution is not harvest unit integer feasible in period t **then**
    Find the most appropriate crew-harvest unit branch
  else
    if Nodal solution is not strategy integer feasible in period t **then**
      Find the most appropriate crew-strategy branch
    end if
  end if
end for

The heuristic for finding branches scans through the nodal solution period by period, from the first period to the last integer period. Branching in this manner is advantageous as the initial periods of the OHS are the most important. The branching heuristic is essentially 'greedy' as it makes decisions for the earlier (more important) periods before it moves to subsequent periods. An added bonus of this approach is that if the B&B is halted before it has successfully terminated the earlier periods of the OHS should be integer. This will enable the user to implement some of the current solution.

The crew-harvest unit branches are chosen before crew-strategy branches are examined because, crew-strategy branches include harvest unit information. If a crew-strategy branch was implemented before a crew-harvest unit branch, the crew may move to harvest other harvest units, and render the crew-strategy branch ineffective.

8.3.2.1 Choice of crew-harvest unit branch

When the nodal solution is analysed to determine if it is harvest unit integer feasible, a table with the current crew-harvest unit allocations is recorded (σ_{ch}) for the first non-integer period ($t = t_L$). Collecting the data for this table is not straightforward, as a restricted movement solution will allow a crew schedule to contribute to more then one allocation in a single period.

$$\sigma_{ch} = \sum_{i:\ H(i,t)=h} x_{ci} + \sum_{\substack{i:\ H(i,t+1)=h,\\ H(i,t-1)=H(i,t)}} x_{ci} \ldots \forall c, h, t = t_L$$

The value of σ_{ch} is the sum of crew schedules that have movement sequences (see Section 8.3.1.1) that indicate the crew is in harvest unit h in period t_L. The table σ_{ch} is used to decide which crew and harvest unit to allocate in a crew-harvest unit branch.

A desirable OHS will try to minimise the movement of the crews between harvest units, this will reduce the penalties applied, and be easier to implement in the forest. To encourage this behaviour in a solution a user adjustable parameter, epsilon, was introduced where $0 < \epsilon < 1$. This parameter is input to the OHSA and is added to the crew allocations to bias branching to favour branches that allow a crew to continue harvesting without moving. So $\sigma'_{ch} = \sigma_{ch} + \epsilon$ for the harvest unit that was branched on in the previous period (t-1), otherwise $\sigma'_{ch} = \sigma_{ch}$.

One-branches are preferred because the OHSA uses a depth-first search. Therefore, at a nodal solution a crew-harvest unit branch is applied to the highest fractional crew, harvest unit allocation in the period. Therefore, in period t_L choose

$$(c, h) \in \arg\max\{\sigma'_{ch} : \sigma_{ch} < 1\}.$$

8.3.2.2 Harvest unit areas

The harvest unit area constraint (see Section 6.4.2.4) can cause some difficulties when choosing the appropriate crew-harvest unit branch. As the area constraint limits the total area of a harvest unit that can be harvested, it also limits the number of allocated periods that a crew may remain in one place.

The harvest unit area constraint affects the branching decision when a harvest unit is near completion. Because of the definition of the crew-harvest unit branch (see Section 8.3.1.1), if a one-branch is imposed on a harvest unit that was occupied in the previous period the crew is forced to remain in that harvest unit until the end of the period. If there is not enough area left in the harvest unit to sustain the crew's operations for an entire period, the problem becomes infeasible. Therefore, the branch choice algorithm must identify which harvest units are completed in the current period, and branch so that crews will move from these harvest unit. The algorithm must determine which harvest unit the crew will move to in the period and impose a branch to force this movement.

The OHSA must also ensure that a harvest unit branched in the manner above is completed before the crew leaves. If the OHSA does not impose this restriction, it is likely that small areas will remain in harvest units that crews have left. The small areas will be left because the exact determination of the crew movement within a period is controlled by the LP solution process. To guarantee the completion of these units, the harvest area constraint (see Section 6.4.2.4) is modified so that the residual area left in the harvest unit after the current period is forced to zero.

8.3.2.3 Choice of crew-strategy branch

Like the crew-harvest unit branches, a strong one-branch is preferred for an effective crew-strategy branch. A table of strategy allocations is found (ς_{chs}). Finding the values of this table is simpler than the σ_{ch} table if only a single strategy is required,

$$\varsigma_{chs} = \sum_{\substack{i:\ H(i,t)=h,\\ S(i,t)=s}} x_{ci} \ldots \forall c, h, s, t = t_L.$$

If multiple compatible strategies are required, the formula becomes

$$\varsigma_{chk} = \sum_{\substack{i:\ H(i,t)=h,\\ S(i,t)\in Q_k}} x_{ci} \ldots \forall c, h, k, t = t_L.$$

Therefore, for a simple crew strategy branch in period t_L choose

$$(c, h, s) \in \arg\max_{c,h,s}\{\varsigma_{chs} : \varsigma_{chs} < 1\}.$$

For a complementary crew strategy branch in period t_L choose

$$(c, h, k) \in \arg\max_{c,h,k}\{\varsigma_{chk} : \varsigma_{chk} < 1\}.$$

If the crew-harvest unit branch has not been applied to crew c and harvest unit h, first crew-harvest unit branch on c, h in period t_L. A complication does occur when the crew will move between harvest units within the period. In this case, a separate crew-strategy branch is required for each harvest unit.

8.3.3 Implementation of branches

To implement a constraint branch, crew schedules that do not comply with the branch are removed from the problem. Firstly, all the crew schedules that are already in the problem are scanned and those that do not comply with the branch are removed. The banned crew schedules have their status changed to become artificial variables and ZIP removes them with a phase one phase two iteration (Section 5.2.1).

As column generation is employed within the B&B process, the column generator must not generate variables that are banned by the constraint branches at the current node. Here a real strength of the constraint branching methodology is seen. If variable branching was used (see Section 5.2.3) the column generation will continue to generate banned crew schedules. However, the structure of the constraint branches in the OHS (Sections 8.3.1.1 and 8.3.1.2) is equivalent to removing arcs in the shortest path formulation in the column generator. Therefore, the column generator can avoid generating banned crew schedules.

The crew-strategy branches are easily included in the column generation algorithm as the banned cutting strategies for a crew, harvest unit, period combinations are simply not considered in the DP.

Crew-harvest unit branches are more difficult to implement as they need to consider the location of the crew in the previous period, to determine whether a crew, harvest unit, period combination is banned. To determine whether an arc is legal with respect to applied crew-harvest unit branches, an array ψ_{cht} is used. Before any branches are applied, ψ_{cht} is initialised to

$$\psi_{cht} = \begin{cases} 1 \iff \text{combination is legal} \\ 0 \iff \text{combination is illegal.} \end{cases}$$

The illegal combinations occur for various reasons. Perhaps the crew is unavailable in the period, therefore, all harvest unit combinations for that crew and period are illegal. When a branch $J_1^H(cht)$ is applied $\psi_{c\hat{h}t} \neq 1$ for $\hat{h} \neq h$. When $J_0^H(cht)$ is applied $\psi_{cht} \neq 1$ for the harvest unit h. In practice, the value of ψ_{cht} is changed to the branch identifier.

To determine if an arc $l_{ch'ht}$ is legal with respect to applied branches three cases are considered.

- $\psi_{ch'(t-1)} = 1$ and $\psi_{cht} = 1$, both the current and previous harvest units (h,h') are legal in their respective periods.
- $\psi_{ch'(t-1)} \neq 1$ and $\psi_{ch(t-1)} = 1$, the previous harvest unit is illegal in period t-1 but the current harvest unit is legal in that period.
- $\psi_{cht} \neq 1$ and $h' = h$, the current harvest unit is illegal but it is the same as the previous harvest unit.

The last two cases allow the crew schedules to create a relaxed integer solution. When an arc ($l_{ch'ht}$) does not fall under any of these cases it is illegal and is not considered in the DP recursion.

8.3.4 Motivation for additional column generation

In Section 8.2.4, we mention some special cases where infeasibility is created after a crew-harvest unit branch. The solution to this problem is to generate additional columns as described in Section 8.2.5. As the infeasibility results from a combination of crew-harvest unit branching and the special treatment of harvest unit area discussed in Section 8.3.2.2, the detailed discussion of the special cases was postponed until this section.

In general, the column generation algorithm in the thesis cannot always generate the required integer feasible columns if area must be conserved in previous periods. To illustrate this an example is given below.

In this example, the OHSA has partially completed the B&B process (Section 8.3), and first non-integer period (t_L) is period six. The initial harvest unit for the crew is **A**. The sequence of previously applied one-branches is A1, B2, B3, B4 and B5 where the letter indicates the harvest unit chosen and the number is the period. Three non-zero variables for a single crew in the current solution are shown in Table 8.5, along with their value in the current solution (x_{ci}). From these values, the allocation for the crew to harvest unit **B** may be calculated. This allocation may be interpreted as the fraction of the period the crew spends in **B** in an unrestricted movement solution. The residual area for harvest unit **B** at the end of the period is also displayed.

Table 8.5: Failing of column generation

	Existing Variables			Harvest unit **B**		Required Variable
Period	V_1	V_2	V_3	Allocation	Residual Area	V^*
x_{ci}	0.79	0.06	0.15			
1	A	A	A	0.00	10.7000	A
2	A	B	A	0.06	10.5158	A
3	B	B	B	1.00	7.2187	B
4	B	B	B	1.00	3.8581	B
5	B	B	B	1.00	0.4975	B
6	C	C	B	0.15	0.0000	B
7	C	C	D	0.00	0.0000	C
8	C	C	D	0.00	0.0000	C

In period six, the branch choice algorithm given in Section 8.3.2 will determine that the crew will finish harvest unit **B** in the current period. It will then force the residual area of **B** to zero in period six, and impose a one crew-harvest unit branch on harvest unit **C** as this harvest unit will have the highest value of $\sigma_{ch} = 0.85$. This branch will make crew schedule V_3 infeasible. The crew schedule required to give a feasible solution after this branch is V^* at value 0.15.

Unfortunately, if the dual variables indicate that entering **B** in period two will give a lower cost to the DP recursion (d_{ht}) in period two. The column generation will not generate V^* as this crew schedule does not enter **B** until period 3. The solution to this problem is discussed in Section 8.2.5 and if $V^* \in N(V_2)$ the crew schedule V^* will already be available to PEVAR and will not need to be generated.

8.4 Integer allocation

In Section 5.2.3.2, integer allocation algorithms are discussed. The integer allocation in ALLOC quickly and efficiently finds good quality solutions to the OHS problem. The B&B can use the objective value of the integer allocation solution as an upper bound for the minimisation problem.

8.4.1 Description

In the integer allocation, the nodal solution is examined period by period as in the B&B. In each period however, crew-harvest unit branches on all of the crews are made simultaneously, then the problem is resolved. Crew-strategy branches are then imposed on all of the crews and the problem is resolved. This process is repeated until all the required periods are integer feasible or branches force the problem to become infeasible.

The integer allocation described above uses the simplex algorithm. This requires significant computational time, however, the quality of the solutions produced does justify this approach. In addition, the large number of linear constraints would make other approaches difficult.

8.4.2 Detailed process

The detailed integer allocation algorithm is given in Algorithm 8.4. Some notes on specific aspects of this algorithm follow.

Algorithm 8.4 Integer allocation algorithm

Save all candidate node information
for each period t **do**
  Rank the crews by the maximum harvest unit allocations in t $R(i_c)$
  for each i_c in $R(i_c)$ **do** $\{$Find stand allocations$\}$
    Find a suitable crew-harvest unit branch
  end for
  Implement a crew-harvest unit branch for each crew
  Resolve the problem
  Regenerate yield predictions
  Resolve the problem
  for each crew **do** $\{$Find the strategy allocations$\}$
    Find crew-strategy branch to implement $\{$Note, two are needed if the crew moves$\}$
  end for
  Implement the crew-strategy branches
  Resolve the problem
  Regenerate yield predictions
  Resolve the problem
end for
if a relaxed integer feasible solution is found **then**
  Report the new upper bound and save the solution
else
  Discard the solution
end if
Restore the original nodal solution and continue branch and bound

As all of the crews are allocated simultaneously there is a possibility that several crews could be allocated into the same harvest unit. In the problem formulation, the number of crews harvesting in a single harvest unit may be restricted. If too many crews are allocated to a harvest unit, the problem becomes infeasible. The crews are allocated to a harvest unit based on the values of σ_{ch} (see Section 8.3.2.1). The highest values of σ_{ch} are allocated first. The number of crews allocated to each harvest unit is tracked. If the harvest unit capacity constraint is exceeded for a particular harvest unit, successive crews indicated for that harvest unit will be allocated to the second highest σ_{ch} value.

When each of the branches are found and implemented, the problem is re-solved and then yield predictions are re-generated as discussed in Section 8.5. This regeneration allows yields to be generated that specifically fit the integer solution of the OHS problem.

8.4.3 Selective use of integer allocation

The integer allocation process yields good quality solutions, this must be balanced against the computational expense of finding these solutions. The integer allocation process in some respects will make similar decisions to the B&B and successive integer allocations for several nodes of the B&B tree can be identical, if the B&B algorithm makes the same decisions as the integer allocation did. To prevent this wasted effort the integer allocation is only attempted in nodes that are expected to generate a new integer solution. This can be achieved if the integer allocation is attempted only on the root node of the B&B tree and subsequently, only after a zero-branch has been made.

8.5 Yield generation

If the OHS problem is solved using a single set of yield predictions the ability to generate new yield predictions as discussed in Section 3.3.1 is ignored.

In any particular harvest unit, yield predictions can be altered to produce better solutions to the OHS. Epstein et al. (1999b) shows significant gains in solution value if an iterative process is used. The literature discusses a number of methods that alter the yield predictions of individual harvest units in a forest level problem (see Section 4.2.1), notably Epstein et al. (1999b), Laroze (1999) and Cossens (1996). Because of the large number of possible alternative yields from a harvest unit, only simple problems can be solved close to true optimality with a limited number of static yields. In the research described in this thesis, a yield generation procedure similar to that found in Cossens (1996) is used.

8.5.1 Purpose and use of yield prediction

Yield generation allows harvesting crews to produce different proportions of log-types from a single harvest unit. As yield generation increases the number of possible crew schedules, the process can only improve the value of the optimal solution compared to static yield optimisation.

The increase in value of a solution that uses yield generation over a solution that does not can vary depending on the initial choice of yields. In the AC case study (see Section 9.5), yield generation was required to find a solution within a reasonable time.

The data in Figure 8.4 and Tables 8.6–8.8 were prepared using the AC case study data. Only the initial log-stocks were downgraded.

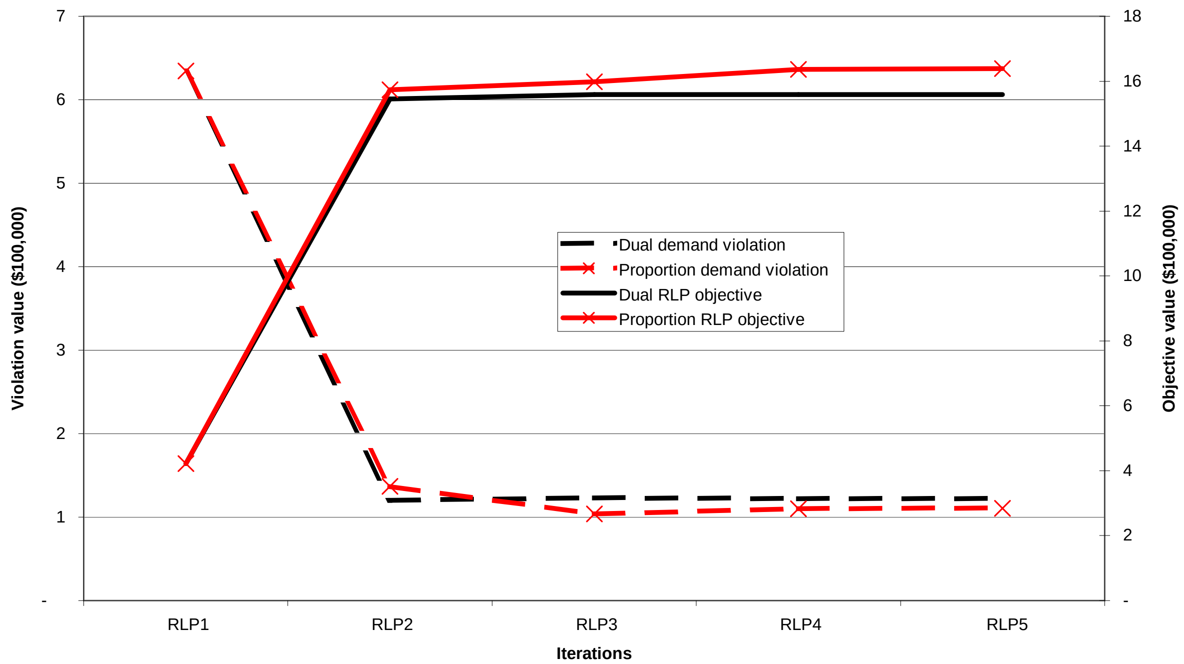


Figure 8.4: A comparison of the two different yield iteration strategies

Figure 8.4 shows the improvement in solution value and the decrease in demand violation when yield generation is used. It must be noted that the major improvement is found in the first iteration of yield generation. This graph is very similar to the graph in Epstein et al. (1999b) that is shown in Figure 4.2.

8.5.2 Implementation issues

The generation of the yield predictions is accomplished outside the OHSA by a commercial software product, MARVL3.5 (Deadman & Goulding 1979). MARVL is used in command line mode to generate the yield predictions. If yield predictions were produced internally, they would be produced within the column generation algorithm. However, the overhead and difficulties involved with calling an external program place the MARVL call outside of PRIMAL. The total number of yield generation iterations is reduced and can be limited by the user.

8.5.3 Initial generation of yield predictions

The initial yield predictions are generated at the beginning of the OHSA. These yield predictions are generated using the roadside sales price for the logs (see Section 2.6.4). These prices are generated by using the delivered sales price for the logs and then subtracting the appropriate transportation cost from harvest unit to customer. The roadside price is used so that harvest units are directed to produce logs for the customers that are located nearby. The approach is significantly different from Cossens (1996), which used stumpage sales prices. However, the OHS formulation in this thesis includes harvesting costs where Cossens (1996) did not.

The subset of logs included in the initial cutting strategies (see Section 6.4.3) are generated using a technique suggested by Glen Murphy (personal communication Murphy, 2001) that ensures each strategy has a balance of high and low quality logs.

8.5.4 Iterative generation of yield predictions

In an iterative algorithm, the proportions of the logs in the yield predictions alter during the progress of the OHSA. In the literature (see Section 4.2.1) there are three methods used to iteratively change the yield predictions.

- Changing the log-types that are included in the cutting strategies. This is described in Epstein et al. (1999b)
- Changing the specifications of log-types to alter the proportions. This is described in Laroze (1999)
- Changing the weighting between log-types in the cutting strategy, either by changing prices in a DP or by changing the priority order in a priority bucker. This is described in Eng et al. (1986), Cossens (1996) and Epstein et al. (1999b)

The approach taken in the OHSA is a progression from the techniques in Cossens (1996) where the changes in yield predictions are directed by altering the prices given to a DP bucker.

The price change approach was chosen as it was the most appropriate approach to use with MARVL. The iterative decomposition of the problem to generate new yield predictions is analogous to a Dantzig-Wolfe price directed decomposition (Dantzig & Wolfe 1960). In this methodology, the master problem uses transfer prices to direct the sub-problems. The transfer prices are derived from the dual variables of the master problem. The master problem is the formulation described in Chapter 6 and the sub-problems use MARVL to determine the yield for each of the harvest units.

Unfortunately, as OHS cutting strategies contain a subset of the log-types, a true optimum solution will not always be found. To find the true optimum the cutting strategy for each harvest unit could be the complete set of logs, however, harvesting crews will not be able to cut this many log-types (see Section 2.5.1.4). Alternatively, some method similar to that in Epstein et al. (1999b) could be used to iteratively generate cutting strategies.

Some alternative methods that can be used to generate transfer prices are discussed below.

8.5.4.1 Dual variables

The transfer prices given to the sub problems in a Dantzig-Wolfe decomposition reflect the change in objective of the master problem as the resource is increased,

$$\hat{c}_r = c_r + \sum_i \pi_i a_{i,r} \tag{8.7}$$

where:

$\hat{c}_r$	is the transfer price of resource r;
c_r	is the objective cost of r in the master problem;
π_i	shadow price of constraint i;
$a_{i,r}$	activity of r on constraint i.

In the OHS problem, the resources are the volume of logs produced in a harvest unit in a period. The sub-problems use MARVL to calculate these volumes as a yield prediction. The yield predictions are then incorporated into crew schedules (see Section 6.7).

As log volumes can be produced from every harvest unit, there should be a transfer price for every log in every harvest unit in every period c_{hlt}.

In the OHS formulation, the revenue from the log sales is calculated from the volume allocation variables (v_{hmlt}, see Section 6.5.1) that allocate logs to a customer, not the crew schedules. The cost of harvesting is constant across all logs harvested by the same crew in a harvest unit and therefore does not affect the relative values. Therefore, $c_r = 0$ in all cases.

The volume allocation constraint (see Section 6.5.1) is the only constraint where the log production from crew schedules is active. The activity of the crew schedules in this constraint is simply a multiple of their proportion in the yield prediction.

Transfer prices are therefore equal to the dual variables on the production constraint,

$$c_{hlt} = \pi^{P}_{hlt} \tag{8.8}$$

where:

c_{hlt}	is the transfer price of the log l in stand h and period t;
π^{P}_{hlt}	shadow price of the volume allocation constraint for log l in stand h and period t.

However, to reduce external calls from the OHSA, yield predictions are only generated from the first period transfer prices (c_{hl1}). The yield predictions generated from these prices tend to provide good solutions as the demands from later periods can affect the volume allocation constraint in the first by the action of the inventory variables (see Section 6.5.5)

Table 8.6: Comparison number of calls to the column generator in the RLP solution (RLP1 has a single call, RLP5 has 5 calls)

	RLP1	RLP2	RLP3	RLP4	RLP5
RLP Objective	422,262	1,544,725	1,558,854	1,559,187	1,559,454
Objective	323,882	1,516,860	1,508,330	1,542,670	1,534,520
Objective without penalty ($)	1,724,833	1,907,475	1,935,243	1,955,216	1,955,558
Value of demand violation ($)	634,360	118,488	125,331	121,938	123,472
Bound gap (%)	23.3	1.8	3.24	1.06	1.6
Solve time (secs)	420.8	78.08	104.77	90.63	120.03
Yield generation time (secs)	75.03	158.36	217.85	289.34	360.98

Table 8.6 compares solutions against the number of iterations of yield generation. In these scenarios, crew-cutting strategy branches were not used. The RLP1 scenario used the at-roadside prices for all log-types (see Section 2.6.4), the algorithm was terminated early because no progress was made towards finding a solution within the bound-gap. Note, that the RLP solution value seems to improve very slowly after the first yield generation RLP2. However, the

RLP objective value has not converged after 5 iterations.

8.5.4.2 Change by proportion

In the research, it was decided to test an alternative method of transfer price generation. This method incorporates ideas on stabilisation of column generation schemes that can be found in du Merle et al. (1999). The principle behind this technique is that the dual variables obtained early in the optimisation are not good indicators of their values in an optimal solution.

This method prevents large changes in the transfer prices by retaining some sense of the original prices and subtly incorporating the over and under supply. To achieve this aim, the original prices were proportionally altered based on the dual variables of the volume allocation constraints. Define Δ as the proportion change such that $0 < \Delta < 1$. The transfer prices can then be found by

$$\hat{c}_{hlt} = \begin{cases} c_{hlt}(1+\Delta) \iff \pi^P_{hlt} < 0 \\ c_{hlt}(1-\Delta) \iff \pi^P_{hlt} > 0 \end{cases} \tag{8.9}$$

where:

$\hat{c}_{hlt}$ is the new transfer price;
Δ is the proportion change.

Unfortunately, the prices for some log-types that are under-supplied can in fact become arbitrarily large in this approach. This is because the price will continue to increase if demand is not met. The prices, though large, are still valid input for the DP bucker but they do not reflect any real information. It is very difficult to get sensible price information (see Section 3.3.3) if the prices are unrealistic. The results of this type of pricing are shown in Table 8.7.

If we compare the RLP objective values shown in Table 8.6 with Table 8.7 we see that changing prices by proportion leads to better RLP solution values (3.5% improvement between the RLP5 and RLP5P solutions), though again there is no convergence shown in Table 8.7. The increase in objective values obtained by this method was also shown in the other case study scenarios. Therefore, this method is used to generate the case study information in Chapter 9. Three yield iterations before the branch and bound is used in all cases.

Table 8.7: Solutions that use yield generation with proportional pricing (RLP1P has a single call, RLP5P has 5 calls)

	RLP1P	RLP2P	RLP3P	RLP4P	RLP5P
RLP Objective	422,262	1,573,189	1,610,007	1,613,301	1,614,336
Objective	323,882	1,515,138	1,572,695	1,593,402	1,555,904
Objective without penalty ($)	1,724,833	1,949,826	1,907,019	1,928,809	1,904,128
Value of demand violation ($)	634,360	136,638	104,313	104,819	106,480
Bound gap (%)	23.3	3.69	2.32	1.23	3.62
Solve time (secs)	420.8	104.11	151.09	136.73	201.3
Yield generation time (secs)	75.03	147.59	222.87	289.63	408.15

8.5.4.3 Pulp wood pricing

In practice, ether of the two pricing methods may lead to the removal ($c_{hlt} = 0$) of over supplied low-grade log-types (i.e., pulp) from yield predictions. In forestry, pulp is the lowest grade of wood quality. Stem sections that are allocated to any other log product may be downgraded to pulp. Without pulp in the yield prediction, the level of waste increases as the volume that should be converted to pulp is wasted. To ensure that the level of wastage is acceptable, pulp is included in every cutting strategy and its price is never allowed to reduce below its initial levels.

8.5.5 Iterative generation of yield in the Branch and Bound

The generation of yield predictions can continue while the branch and bound process is taking place. This is especially useful when complementary crew-strategy branches are used that restrict the number of log-types harvested by each crew. Once a complementary crew-strategy branch is implemented new yield predictions can be generated that will improve the solution value. This approach is not mentioned in any of the literature examined in this thesis.

Table 8.8 compares solutions against the number of iterations of yield generation within the B&B algorithm, or more specifically within the integer allocation process. In these scenarios, crew-strategy branches were used, as they illustrate the difference made by yield generation in B&B process. However, as the problem is more restricted, the objective values are less than those shown in Tables 8.6 and 8.7.

Table 8.8: Comparison of solutions with number of calls to the column generator in the Integer Allocation

	RLP5Branch1	RLP5Branch2	RLP5Branch3	RLP5Branch4	Strategy
RLP Objective	1,559,454	1,559,454	1,559,454	1,559,454	1,598,726
Objective	1,490,381	1,494,416	1,494,419	1,491,095	1,385,401
Objective without penalty ($)	1,902,418	1,896,833	1,896,831	1,905,493	1,882,402
Value of demand violation ($)	126,835	133,016	133,013	137,049	188,708
Bound gap (%)	4.43	4.17	4.17	4.38	13.34
Solve time (secs)	1,285.78	252.14	267.62	1,096.03	1,206.38
Yield generation time (secs)	441.07	511.29	576.07	647.4	222.87

The cases RLP5Branch1 to RLP5Branch4 use yield iteration five times in the RLP solution process and one to four times in the B&B (it also uses proportional yield generation). The case Strategy (included from Section 9.5.2) uses yield generation three times in the RLP solution and not at all in the B&B. The advantages of yield generation over not using yield generation can be clearly seen. The use of yield generation twice or three times in the B&B seems to give the best performance. However, the behaviour of the RLPBranch4 solution is problematic and perhaps indicates that the integer allocation process may need fine-tuning. It must be remembered that all these problems are solved with a bound gap of 5%, except the Strategy case that has a 10% bound gap.

8.6 Resolution of end-effects

The OHS problem is a continual process. The forest will continue to be harvested into the future. However, the data available at any one time can only model a finite number of periods. The OHS shares this trait with other levels of the hierarchical planning process (Section 4.1). This is because long-term data are unavailable, inaccurate or too expensive to provide. The OHS problem formulation in this thesis, therefore, only models a small part of the continual process of Operational Harvest Scheduling.

Harvest schedulers, in practice, use available data to plan the current period's (one week) harvest. Then at the beginning of each new week a new plan is found. Any future weeks that are planned, only forecast future decisions, which may be changed when the actual plan for that week is implemented. This type of practical planning process leads to *rolling horizon models*.

In a rolling horizon model, an OHS solution will schedule the harvest of a forest over several weeks (or periods). However, a new schedule will be re-solved at the beginning of each week, with new data. In any weekly plan, only the first week's solution is implemented. The rolling horizon is a practical method needed to implement the results of a finite horizon model in a continual process. However, the truncation of the time horizon can lead to problems that affect the results given in weekly solutions. These problems are called *end-effects*. If not controlled end-effects may influence the operational decisions.

8.6.1 Nature of end-effects

The OHS is susceptible to end-effects in three different areas.

- Final period decisions.
- Re-entry costs.
- Shortage of candidate harvest units.

In the final period of the OHS, the decisions in the optimal solution can be very different to decisions that would be made by a rational harvest planner. The model does not consider revenue from decisions that will be made in subsequent unmodelled periods. Therefore, in the last period any options that will generate marginal value will be used. For instance, residual log-stocks of high-value log-types will be downgraded to pulp and sold. A rational harvest planner may decide to keep these log-stocks, because, they will be sold without downgrading in future periods.

If crews leave harvest units before they are clearfelled, a crew will have to return to the unit to finish harvesting, before it can be replanted. If an OHS solution delays returning a crew to the harvest unit until after the end the time horizon, the re-entry costs will not be considered. Therefore, a OHS model may move crews from unfinished harvest units more than a rational harvest planner.

In a multi-period model, near the end of the time horizon, the number of unharvested candidate harvest units may be low. This will reduce the choices available to the model. In practice, with a rolling time horizon, new harvest units will be added to the problem, from the harvest planning process, before the week's solution is implemented. The periods near the end of the time horizon will, therefore, eventually have a greater choice of harvest units than they currently have in the model.

End-effects do not only affect the solution in the final periods, sometimes end-effects can propagate into earlier periods of a multi-period model. For example, because crew re-entry costs are not calculated if a crew returns after the end of the time horizon, the OHS solution will frequently move crews from unfinished harvest units. These movements can occur in any period.

Another example of the propagation of end-effects occurs when strictly integer solutions (Section 7.1) are required. In this type of solution, crews need to move between periods. Productivity and production will be adjusted in preceding periods so that the crew completes harvesting exactly at the end of a period. This implies that constraints on crew movement in a later period will force changes on crew decisions in earlier periods. In a relaxed integer solution this effect does not occur.

8.6.2 Methods of resolution

There are several different ways of reducing the influence of end-effects.

- Extending the time horizon.
- Discounting later periods.
- Adding constraints.
- Using residual costs.
- Use of higher-level plans.

Of these five possible techniques, in this thesis the following are used: extending the time horizon; adding constraints; residual costs.

The first technique commonly used to reduce end-effects, is to extend the time horizon of the model. In a rolling horizon formulation, only the solution for the first period is used. In effect, any other periods in a multi-period model are ‘extra periods’. These extra periods are used to indicate future decisions, but they also serve to reduce the influence of end-effects in the first period. By adding more periods, this technique reduces the influence of end-effects in earlier periods. The OHSA can relax the integer restrictions on the extra weeks and impose no branches on these periods’ solutions. Therefore, these extra periods will not significantly increase the solution time.

In the case studies (see Chapter 9), the results for all the periods given in the case study data were examined. Therefore, extra period information was generated to reduce the observed end-effects in the solution for these periods.

If the objective function coefficients of decisions in later periods of a multi-period solution are discounted, then any end-effects in these periods would have less influence on the total solution value. This is an extension of the extra periods technique above. Some end-effects will not propagate into earlier periods when this technique is used, because earlier decisions will have a greater effect on the objective function. However, constraint driven end-effects, such as the integer crew movement example above, will not be affected by any discounting.

This technique is commonly used in forestry strategic planning models. In the strategic plan, discounting has a practical interpretation as it reflects the present value of revenue obtained in the future. However, since the time horizons in the OHS problem are so short discounting will have no practical interpretation.

In order to reduce aberrant decisions in the final period, extra constraints can be added to the model. These constraints restrict the residual quantities in a solution. If, the minimum residual log-stock volumes were constrained, the model will not be able to downgrade high-value logs in the last period. However, setting the levels of these constraints can be problematic. The problem formulation in this thesis does not constrain final log-stock volumes, however, a constraint can force the residual area of a harvest unit to zero (see Section 6.4.2.5). This ensures that the harvest unit is clearfelled within the time horizon. In addition, in this thesis there is the ability to force a crew to remain in a harvest unit until it is clearfelled. This restriction reduces the re-entry cost end-effect, because crew movements are reduced. Some of the case study scenarios implemented this constraint.

Residual prices value residual quantities of log-stocks or high-value harvest unit areas. If residual prices are used aberrant decisions in the final period are reduced. This technique is similar to the constraint technique mentioned above, but is more flexible because the volumes are not constrained. However, setting the prices can be difficult. A price that is too low will have no influence on behaviour, while a price that is too high, may in fact stop any crews from harvesting a harvest unit (if residual area prices are used), or any downgrade of log-stocks (if residual log-stocks prices are used). In the formulation in the thesis, these prices are considered (Section 6.6.5) but none of the case studies used this ability.

Residual prices for the OHS can be generated by a tactical planning model (see Section 2.7.1). Because the tactical plan has a longer time horizon than the OHS it can correctly value the residual log-stock volumes or harvest unit areas. The tactical plan can be incorporated in two ways. It can be a separate model, its output forming some of the OHS input. Or, the tactical planning model and the OHS model could be integrated into a single variable resolution model and solved together. The second approach was used successfully in McNaughton (1998), where a tactical planning model was linked to a strategic planning model. In this thesis, no explicit information from a tactical plan was used.

8.7 Concise solution strategy

Figure 8.5 shows in a concise diagram the solution strategies used in this thesis.

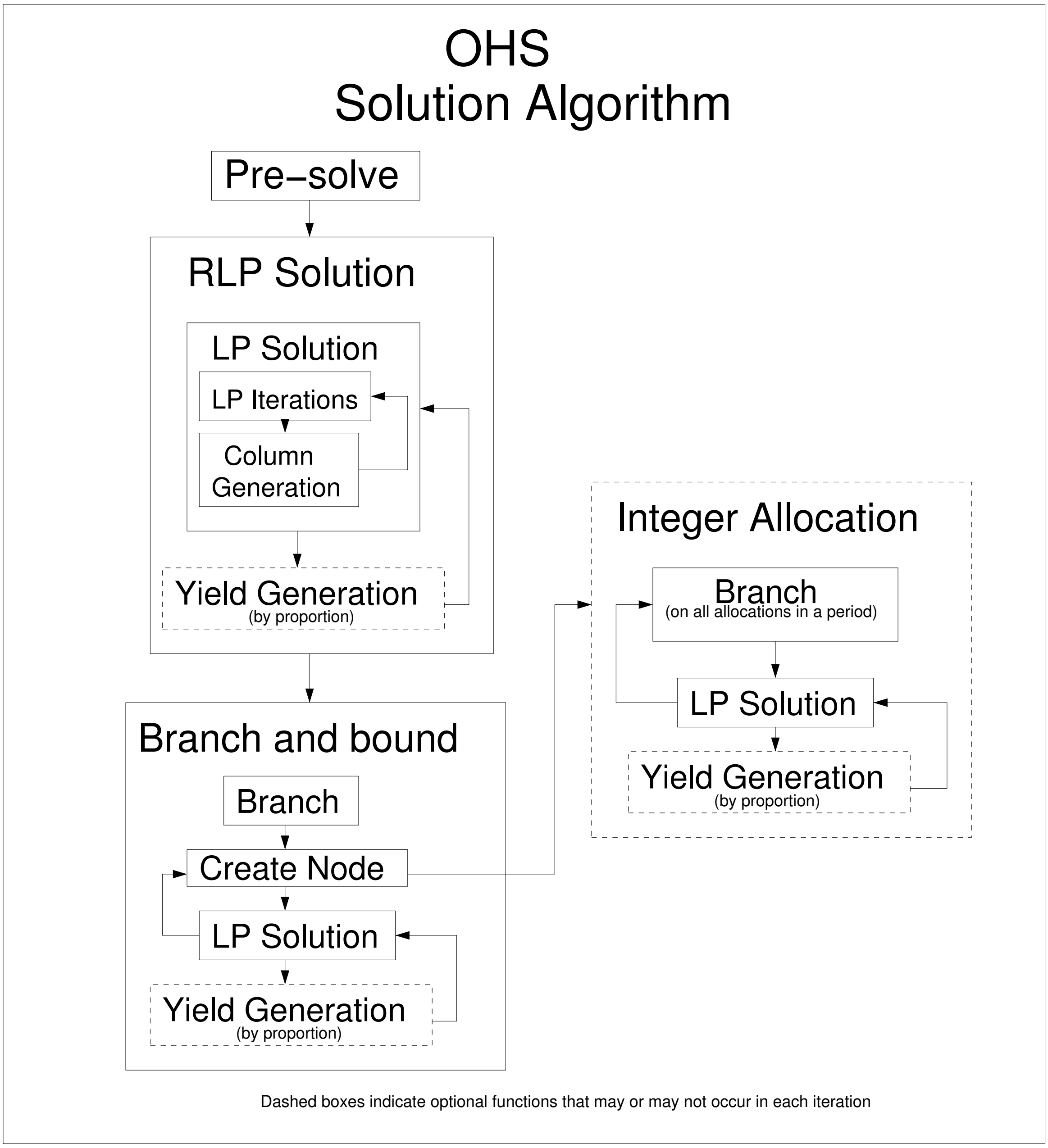


Figure 8.5: A concise diagram of the solution strategy used

Chapter 9

Case Studies

Amid whose swift half-intermitted burst
Huge fragments vaulted like rebounding hail,
Or chaffy grain beneath the thresher's flail:
(Coleridge 1798)

Three case studies for two forestry companies were undertaken during the course of this research. They were:

- the NZC operational schedule (two weeks) case study;
- the NZC annual schedule (twelve months) case study;
- the AC operational schedule (four weeks) case study.

These case studies investigated whether the formulation of the problem was realistic and tested the solution algorithms on reasonably sized problems. These case studies also provided example problems that were used to develop the solution process.

9.1 Introduction

The two companies involved are not named in this thesis because of concerns for commercial sensitivity. In addition, the crew and harvest unit names have been changed. One of the companies examined is based in the central North Island of New Zealand; this company will be referred to as New Zealand Company (NZC). The other company's operations are in southeastern Australia and will be referred to as Australian Company (AC).

The case studies were undertaken at different times during the research. The NZC case studies were undertaken before the AC case study. Thus, the NZC studies do not utilise all the techniques discussed in this thesis. In particular, yield generation data was not collected for NZC so fixed yield predictions are used in the NZC operational and annual schedules.

While compiling these three case studies, I found that the problem definition was subtly different between the two companies. These differences were reflected by the different kinds of data available to the companies and differences in their operations. For instance, NZC assumed constant productivity for each crew, while in the AC data the crew productivity was a function of the type of harvest unit.

These differences changed the input information for the model, and the statement of the PT sub-model (see Section 6.5). The formulation developed for the NZC model was the base model, the AC case study extended this model.

Before the case study data were collected, test data from previous studies at *forest research* were used in the development of the formulation and algorithm. Most of the data were collected for a single period only. In some cases, the data were entirely simulated and were not generated from real forest problems.

The data for these case studies were specifically collected with the requirements and abilities of the OHS model in mind. For example, the demand targets collected were flexible enough to allow the optimisation to work effectively. In the AC study the yield generation base data (inventory assessments) were collected, therefore, yield predictions could be generated within the algorithm.

These case studies emphasised the importance of several features of the solution process. The NZC operational case study, without yield generation, highlighted the importance of yield generation or downgrading. The AC study, highlighted the strategies needed to: incorporate large volumes of log-stocks; include downgrading; calculate the objective.

9.2 New Zealand Company (NZC)

There were two separate case studies provided by NZC at the end of 2000. The first was an operational schedule with a two week planning horizon. This case study included:

- 7 crews;
- 22 log-types;
- 18 harvest units;
- 16 cutting strategies for each harvest unit.

The second data-set was the longer-term annual harvest schedule for NZC. In this schedule, the time horizon was one year, divided into 26 fortnightly periods. The log-types were reduced from 22 log-types in the operational schedule, into 12 aggregate groups. Thirty-nine harvest units were considered and an additional crew was included, as a hauler crew was not allocated in the operational schedule. As the log-types were aggregated, only a single yield prediction was used for each harvest unit. The yield prediction was identical to yield predictions used in longer-term forest plans, specifically the NZC strategic plan (see Section 4.1.1).

9.2.1 Data Collection

At an early stage in the research, Paul Cossens and I held discussions with the staff at NZC. These discussions allowed the OHS formulation to be modified to fit NZC's particular situation. A Microsoft Excel™ workbook was compiled and sent to NZC to collect the necessary information for both case studies. The workbook information was modified slightly to fit the existing model and the data were used to formulate both case studies.

9.3 NZC operational schedule

The operational case study was based on data from two weeks in April 2000. To reduce the influence of end effects (see Section 8.6.1) a third week was added to the problem. The market data for the extra week was a copy of the second week's data and no branching was used. Only one week was added because that was the minimum required to add a non-integer buffer to the end of the case study. The results for the extra week are not included in any of the following results.

This case study used the actual log-types and customer demand for the weeks. Initial log-stocks information (see Section 3.5.2) was not provided, so the solution could only meet demand from production within the time horizon. Only minimum market demand volumes (see Section 6.5.2) were supplied, the maximum volumes were left blank. Therefore, maximum market demand volumes were set at 120% of the minimum volumes.

The yields were pre-generated by NZC using MARVL. Sixteen cutting strategies were used to generate yield predictions. As no yield assessments were collected by *forest research*, the case study was solved with fixed yield predictions, no extra yield generation could take place. No downgrade information was collected for the log-types. Because of the lack of information, the only downgrading allowed was the conversion of all log-types into pulp logs. This downgrade was possible because pulp logs are assumed to have the most inclusive log specification (see Section 2.6).

Initial results showed that it was difficult to find a solution that met demand constraints. The high-value log-types were oversupplied while the pulp log volume was under-supplied. There was also a shortfall in overall volume. Therefore, the scenarios discussed below, were developed to investigate ways to reduce the violation of the demand constraints. The ability to downgrade high-value log-type volume to pulp wood was particularly necessary.

9.3.1 Results

Table 9.1: NZC operational schedule results for a number of demand scenarios

	Manual	Manual_DG	Unrestricted	Demand_Min	Demand	High_Pen	No_DG
RLP Objective	-231,226	338,792	1,966,410	804,862	605,331	-2,832,611	98,552
Objective	-236,913	326,900	1,960,194	794,127	583,776	-2,933,977	78,045
Objective without penalty ($)	1,669,946	1,380,674	1,960,194	1,845,193	1,475,780	1,445,304	1,639,343
Value of demand violation ($)	953,430	526,887	2,032,203	1,023,075	446,002	437,928	780,648
Bound gap (%)	2.46	3.51	0.32	1.33	3.56	3.58	20.81
Solve time (secs)	3.5	3.22	1.48	4.19	8.47	5.75	73.36

The row headings in Table 9.1 are similar to those found in Chapter 8 and are listed below.

- RLP Objective: The objective value of the RLP solution multiplied by -1 to give the solution profit minus any applied penalties;

- Objective: The objective value of the integer solution multiplied by -1 to give the solution profit minus any applied penalties;
- Objective without penalty: The objective value of the integer solution without subtraction of penalty values. This represents the dollar value of the solution;
- Value of demand violation: The volume of shortfall or excess for each log-type times its price.
- Bound gap: The percentage difference between objective values of the best integer solution and the RLP solution.
- Solve time: The time taken in seconds for the algorithm to finish.

The description of the case study scenarios is as follows.

- Manual: The crews remain in the initial harvest units. Maximum and minimum demand restrictions are used.
- Manual_DG: The "Manual" scenario but with the ability to downgrade excess volume to pulp.
- Unrestricted: An optimised scenario where the crews may move to other harvest units. Downgrades are allowed and the maximum and minimum constraints on demand are removed.
- Demand_Min: The "Unrestricted" scenario with minimum demand constraints, the penalty for shortfall was twice the market log price.
- Demand: The "Demand_Min" scenario with maximum demands. The penalty for excess was twice the market log price.
- High_Pen: The "Demand" with the demand penalties set at ten times the market log price.
- No_DG: The "Demand" scenario with the ability to downgrade removed.

In all these scenarios, the bound tolerance (see Section 5.2.3.2) was set to 5%. The cutting strategies were not restricted in any of the scenarios. Therefore, crews were able to use a number of cutting strategies in each period. The implementation of cutting strategy branches is examined in the AC case study (see Section 9.5).

The manual scenarios are created by forcing crews to continue to harvest in the initial harvest units selected by NZC. There is enough area for the crews remain in the harvest units for the entire time horizon and it is assumed that the harvest scheduler would not move the crews unless they had completed a harvest unit. The crew allocations for the "Manual_DG" scenario are shown in Table 9.2, only the allocations for the first two periods are shown. It must be noted, no information on the manual allocation of cutting strategies and volume downgrading was collected. Therefore, in the two manual scenarios, these decisions are optimised by the OHSA. In my opinion, a *true* manual scenario will have a worse objective value.

Table 9.2: NZC “Manual_DG” crew allocations

Crew	Harvest Unit	Strategy	1	2
C1	HU11_C1	HU11_C1-10	1.00	1.00
C2	HU11_C2	HU11_C2-01	0.23	0.24
		HU11_C2-02	0.77	0.76
C3	HU12_C3	HU12_C3-02	0.87	0.20
		HU12_C3-10	0.13	0.25
	HU26	HU26-05		0.55
C4	HU38_C4	HU38_C4-02	0.19	0.61
		HU38_C4-13	0.21	
		HU38_C4-14	0.60	0.39
C5	HU1_C5	HU1_C5-10	1.00	1.00
C6	HU38_C6	HU38_C6-04	0.82	0.68
		HU38_C6-14	0.18	0.32
C7	HU6	HU6-13	0.66	0.48
		HU6-14		0.14
		HU6-15	0.34	0.38

In Table 9.2, the values in the cells indicate the time, in periods, the crew was in a harvest unit using the cutting strategy indicated. Note, the values in each period for a crew sum to one, as they represent the crew allocation. In this table, Crew C3 is the only crew to move harvest units. C3 moves almost half way though the second period. C3 is forced to move because it has completed clearfelling harvest unit HU12_C3. We see this in Table 9.3, which shows the residual area of each harvest unit at the end of each period.

Table 9.4 shows the crew allocations for the “Demand” scenario, which optimises crew allocation.

In the “Demand” scenario solution, only crew C3 moves during the time horizon. C3 moves at the same time as it does in the “Manual_DG” scenario for the same reason. However, C2, C5, and C1 have all been allocated to harvest units other than their initial placement. This indicates that these crews should move to the new harvest units before the first period.

Table 9.3: NZC "Manual" harvest unit areas

Stand	1	2
HU1_C5	9.11	7.61
HU6	17.33	15.16
HU11_C2	12.39	10.37
HU11_C1	11.88	9.46
HU12_C3	1.18	0.00
HU26	23.90	22.36
HU37	31.30	31.30
HU38_C4	3.41	0.41
HU38_C6	15.27	12.04

Table 9.4: NZC "Demand" crew allocations

Crew	Harvest Unit	Strategy	1	2
C1	HU26	HU26-05	0.74	1.00
		HU26-07	0.26	
C2	HU27	HU27-02	0.52	0.48
		HU27-10	0.48	0.52
C3	HU12_C3	HU12_C3-01	0.59	0.41
		HU12_C3-02	0.41	0.04
	HU31	HU31-10		0.55
C4	HU38_C4	HU38_C4-06	0.38	0.49
		HU38_C4-10	0.62	0.51
C5	HU27	HU27-02	0.27	0.29
		HU27-10	0.73	0.71
C6	HU38_C6	HU38_C6-02		0.34
		HU38_C6-14	1.00	0.66
C7	HU6	HU6-10	0.51	0.40
		HU6-13	0.00	
		HU6-15	0.49	0.60

9.3.2 Discussion

The most striking result from Table 9.1 is that none of the scenarios can meet the demand constraints. The log-type volumes (not shown because of commercial sensitivity), for the scenarios without downgrading, indicate that high-value log-types are oversupplied while low-value log-types are under-supplied. The minimum value of the demand violation, found in the High_Pen scenario, is very high, 30% of the objective value without penalties. This result is similar to those in Murphy (1998) and is caused by:

- a lack of productive capacity in the case study;

- the inability to change yield predictions to meet the demand constraints.

The lack of capacity indicates that the perhaps the case study data for crew productivity and demand is incorrect. Alternatively, NZC may have supplied *forest research* with optimistic demand figures that are impossible for them to meet operationally.

In Table 9.1, we see the best (minimum) value of demand violation is given by the “High_Pen” scenario. Increasing the penalties for demand violation above 2 times the market log price has little effect, however, as this figure is only 2% better then the “Demand” scenario.

Optimisation alone can improve (reduce) the amount of demand violation, this is seen when we compare the scenarios without downgrades. The “No_DG” scenario is 18% better than the “Manual” scenario.

Downgrading excess volume to pulp dramatically improves the demand violation, “Manual_DG” scenario is 44% better than the “Manual” scenario and the “Demand” scenario is 42% better than the “No_DG” scenario. This effect is due to the downgrade of excess high-value log-types. These results taken from Table 9.1 are summarised in Table 9.5. Note that there is no iterative yield generation in these scenarios.

Table 9.5: NZC Comparison of demand violations with crew allocation and downgrades

	Crew Allocations	
	Manual	Optimised
No Downgrades	953,430	780,648
Downgrades	526,887	446,002

There is significant difference between the unconstrained maximum value of logs in the forest and the maximum value of logs that can be sold to the current market. The objective value without penalties of the “Demand” scenario is 25% worse (less) than the “Unrestricted” scenario.

The overall improvement from optimisation can be seen by a comparison of the “Demand” and the “Manual_DG” scenarios. The “Demand” scenario has an improvement (increase) of 7% in the objective value without penalties, and a 15% improvement in the value of demand violation.

The bound gaps are acceptable (except “No_DG” scenario) given the large possible errors in the underlying yield predictions (see Section 3.3.2).

The solution times for these scenarios were short. When downgrades are allowed between the log-types, the solution process is made significantly easier. This is clearly shown by the solution time increase when downgrades are removed in the “No_DG” scenario. The solution time for the “Manual” scenario is short, even though downgrades are not allowed, because of the more restrictive nature of this scenario.

Cutting strategy branches were not imposed in this case study. It was difficult to find solutions that were near the demand constraints. If the cutting strategy choices were restricted, it would be almost impossible to find solutions. The operation of crew-strategy branches is examined in the AC case study.

9.4 NZC annual schedule

The NZC annual schedule ensures continuity of harvesting over the entire year. It is similar in some respects to a tactical plan (see Section 4.1.2), however, it does not have the extra spatial restrictions common to tactical plans. In the annual schedule, log-types were aggregated into the groupings used in the NZC strategic planning models (see Section 4.1.1). Therefore, the demand constraints were not as restrictive as in the short-term operational schedule. In fact, only 3 out of the 12 log-types had minimum demand constraints and no log-types had maximum demand constraints.

As the annual schedule had a much longer time horizon than the operational schedule, the crews moved between harvest units more often. The techniques developed to solve this problem are discussed in Chapter 8. In particular, the concept and implementation of integer feasible columns (see Section 7.2.1) was necessary to reduce the gap between the RLP and integer objectives.

9.4.1 Results

Table 9.6 shows the results for three scenarios for the NZC annual schedule. The rows of the table are identical to Table 8.1. No manual solutions were simulated as the manual solution data for this case study was not collected from NZC.

- Base: This scenario optimises the annual schedule without downgrading and with penalties of 2 times the market log-price for violation of the demand constraints.

- High_Pen: This scenario applies a higher penalty to the demand constraints.
- Force_Crews: This scenario investigates the effect if the OHSA does not allow a crew to move before it has completed its current harvest unit.

Table 9.6: NZC annual schedule case study results

	Base	High_Pen	Force_Crews
RLP Objective	19,175,130	9,789,744	18,292,376
Objective	19,156,934	9,772,206	18,180,609
Objective without penalty ($)	19,623,611	18,834,901	18,681,075
Value of demand violation ($)	233,338	181,253	250,233
Bound gap (%)	0.09	0.18	0.61
Solve time (secs)	77.95	260.11	159.64

The increase in the number of periods in the annual schedule, is reflected by an increase in the solution times compared to the operational schedule. Fortunately, the use of integer feasible columns has resulted in bound gaps that are smaller than the bound gaps in the operational schedule. The crew allocations for the "Base" scenario are shown in Table 9.7 and the residual areas in Table 9.8.

In Table 9.7, we see that crew C7 moves in and out of harvest unit HU14 during the time horizon and completes a number of different harvest units before returning to HU14. This behaviour is very odd for a hauler crew, as hauler crew's rarely move before they complete a harvest unit. Therefore, the solution may need to be altered manually to force C7 to remain in a single harvest unite, or the cost of moving increased to further discourage this movement.

If we restrict movement of the crews, as discussed in Section 8.6.2, we get the allocations given in Table 9.9. This table shows the allocations for the "Force_Crews" scenario. In this scenario, once a crew enters a harvest unit the crew must remain until it is completed. Therefore, crew movement is reduced compared to Table 9.7.

Table 9.7: NZC annual schedule “Base” crew allocations

Crew	Harvest Unit	1	2	3	4	5	6	7	8	9	10	11	12	13
C1	HU22	1.00	1.00	1.00	1.00	1.0 0	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1. 00
	HU39													
C2	HU27							0.59	1.00	1.00	1.00	1.00	1.00	1.00
	HU40	1.00	1.00	1.00	1.00	1.00	1.00	0.41						
C3	HU8	0.06												
	HU36	0.94	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	HU42													
C4	HU26													
	HU37	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
C5	HU23													
	HU41	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
C6	HU36												0.96	1.00
	HU37													
	HU39	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.04	
C7	HU14	1.00	0.36				0.65	1.00	0.08					
	HU18								0.92	1.00	1.00	1.00	1.00	1.00
	HU21		0.64	1.00	1.00	1.00	0.35							
	HU25													
C1_HL	HU30							1.00	1.00	1.00	1.00	1.00	1.00	1.00

Crew	Harvest Unit	13	14	15	16	17	18	19	20	21	22	23	24
C1	HU22	1.00	0.62										
	HU39		0.38	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
C2	HU27	1.00	1.00	1.00	1.00	1. 00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	HU40												
C3	HU8												
	HU36	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.57
	HU42												0.43
C4	HU26						1.00	1.00	1.00	1.00	1.00	1.00	1.00
	HU37	1.00	1.00	1.00	1.00	1.00							
C5	HU23												0.28
	HU41	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.72
C6	HU36	1.00	1.00	1.00	1.00	1. 00	1.00	1.00	1.00	1.00	1.00		
	HU37											1.00	1.00
	HU39												
C7	HU14							0.65	1.00	1.00	1.00	1.00	0.56
	HU18	1.00	1.00	1.00	1.00	1.00	1.00	0.35					
	HU21												
	HU25												0.44
C1_HL	HU30	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00

Table 9.8: NZC annual schedule “Base” harvest unit areas

Stand	1	2	3	4	5	6	7	8	9	10	11	12	13
HU6	47.10	47.10	47.10	47.10	47.10	47. 10	47.10	47.10	47.10	47.10	47.10	47.10	47.10
HU8	22.17	22.17	22.17	22.17	22.17	22 .17	22.17	22.17	22.17	22.17	22.17	22.17	22.17
HU14	29.13	28.07	28.07	28.07	28.07	26. 20	23.27	23.03	23.03	23.03	23.03	23.03	23.03
HU17	42.70	42.70	42.70	42.70	42.70	42. 70	42.70	42.70	42.70	42.70	42.70	42.70	42.70
HU18	33.50	33.50	33.50	33.50	33.50	33. 50	33.50	30.85	27.88	24.90	21.91	18.93	15.95
HU20	62.80	62.80	62.80	62.80	62.80	62. 80	62.80	62.80	62.80	62.80	62.80	62.80	62.80
HU21	9.10	7.67	5.40	3.10	0.80	0. 00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
HU22	35.49	32.68	29.86	27.05	24.24	21. 43	18.61	15.80	12.99	10.18	7.36	4.55	1.74
HU23	6.30	6.30	6.30	6.30	6.30	6. 30	6.30	6.30	6.30	6.30	6.30	6.30	6.30
HU25	7.10	7.10	7.10	7.10	7.10	7. 10	7.10	7.10	7.10	7.10	7.10	7.10	7.10
HU26	23.90	23.90	23.90	23.90	23.90	23. 90	23.90	23.90	23.90	23.90	23.90	23.90	23.90
HU27	51.80	51.80	51.80	51.80	51.80	51. 80	50.35	47.83	45.25	42.68	40.10	37.53	34.95
HU28	51.70	51.70	51.70	51.70	51.70	51. 70	51.70	51.70	51.70	51.70	51.70	51.70	51.70
HU30	44.00	44.00	44.00	44.00	44.00	44. 00	41.79	39.50	37.21	34.91	32.62	30.32	28.03
HU33	10.40	10.40	10.40	10.40	10.40	10. 40	10.40	10.40	10.40	10.40	10.40	10.40	10.40
HU36	84.11	81.60	79.09	76.58	74.07	71. 56	69.04	66.53	64.02	61.51	59.00	54.16	49.14
HU37	61.87	58.75	55.62	52.50	49.37	46. 24	43.12	39.99	36.87	33.74	30.61	27.49	24.36
HU39	47.16	45.05	42.95	40.84	38.73	36. 62	34.52	32.41	30.30	28.20	26.09	26.01	26.01
HU40	9.09	7.41	5.73	4.05	2.37	0. 69	0.00	0.00	0.00	0.00	0.00	0.00	0.00
HU41	88.49	85.65	82.80	79.95	77.10	74. 25	71.41	68.56	65.71	62.86	60.01	57.17	54.32
HU42	9.30	9.30	9.30	9.30	9.30	9. 30	9.30	9.30	9.30	9.30	9.30	9.30	9.30
HU43	8.50	8.50	8.50	8.50	8.50	8. 50	8.50	8.50	8.50	8.50	8.50	8.50	8.50

Stand	13	14	15	16	17	18	19	20	21	22	23	24
HU6	47.10	47.10	47.10	47.10	47.10	47.1 0	47.10	47.10	47.10	47.10	47.10	47.10
HU8	22.17	22.17	22.17	22.17	22.17	22. 17	22.17	22.17	22.17	22.17	22.17	22.17
HU14	23.03	23.03	23.03	23.03	23.03	23.0 3	21.15	18.22	15.25	12.28	9.32	7.65
HU17	42.70	42.70	42.70	42.70	42.70	42.7 0	42.70	42.70	42.70	42.70	42.70	42.70
HU18	15.95	12.96	9.98	7.00	4.02	1.0 3	0.00	0.00	0.00	0.00	0.00	0.00
HU20	62.80	62.80	62.80	62.80	62.80	62.8 0	62.80	62.80	62.80	62.80	62.80	62.80
HU21	0.00	0.00	0.00	0.00	0.00	0.0 0	0.00	0.00	0.00	0.00	0.00	0.00
HU22	1.74	0.00	0.00	0.00	0.00	0.0 0	0.00	0.00	0.00	0.00	0.00	0.00
HU23	6.30	6.30	6.30	6.30	6.30	6.3 0	6.30	6.30	6.30	6.30	6.30	5.54
HU25	7.10	7.10	7.10	7.10	7.10	7.1 0	7.10	7.10	7.10	7.10	7.10	5.85
HU26	23.90	23.90	23.90	23.90	23.90	21.3 3	18.66	15.99	13.33	10.66	7.99	5.32
HU27	34.95	32.38	29.80	27.23	24.65	22.0 8	19.50	16.93	14.35	11.78	9.20	6.63
HU28	51.70	51.70	51.70	51.70	51.70	51.7 0	51.70	51.70	51.70	51.70	51.70	51.70
HU30	28.03	25.74	23.44	21.15	18.86	16.5 6	14.27	11.98	9.68	7.39	5.10	2.80
HU33	10.40	10.40	10.40	10.40	10.40	10.4 0	10.40	10.40	10.40	10.40	10.40	10.40
HU36	49.14	44.12	39.09	34.07	29.05	24.0 2	19.00	13.98	8.95	3.93	1.42	0.00
HU37	24.36	21.24	18.11	14.99	11.86	11.8 6	11.86	11.86	11.86	11.86	8.85	5.72
HU39	26.01	25.24	23.18	21.07	18.97	16.8 6	14.75	12.64	10.54	8.43	6.32	4.21
HU40	0.00	0.00	0.00	0.00	0.00	0.0 0	0.00	0.00	0.00	0.00	0.00	0.00
HU41	54.32	51.47	48.62	45.77	42.93	40.0 8	37.23	34.38	31.53	28.69	25.84	23.80
HU42	9.30	9.30	9.30	9.30	9.30	9.3 0	9.30	9.30	9.30	9.30	9.30	7.68
HU43	8.50	8.50	8.50	8.50	8.50	8.5 0	8.50	8.50	8.50	8.50	8.50	8.50

Table 9.9: NZC annual schedule "Force_Crews" crew allocations

Crew	Harvest Unit	1	2	3	4	5	6	7	8	9	10	11	12	13
C1	HU22	1.00	1.00	1.00	1.00	1.0 0	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1. 00
	HU26													
	HU42													
C2	HU39				0.11	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	HU43	1.00	1.00	1.00	0.89									
C3	HU8	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.75			
	HU36										0.25	1.00	1.00	1 .00
C4	HU21													
	HU37	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
C5	HU20	1.00	1.00	1.00	1.00	1. 00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1 .00
C6	HU17	1.00	1.00	1.00	1.00	1 .00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	HU36													
C7	HU6	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	HU25													
C1_HL	HU30							1.00	1.00	1.00	1.00	1.00	1.00	1.00

Crew	Harvest Unit	13	14	15	16	17	18	19	20	21	22	23	24
C1	HU22	1.00	0.62										
	HU26					0.95	1.00	1.00	1.00	1. 00	1.00	1.00	1.00
	HU42		0.38	1.00	1.00	0.05							
C2	HU39	1.00	1.00	1.00	1.00	1. 00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	HU43												
C3	HU8												
	HU36	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
C4	HU21									0.21	1.00	1. 00	1.00
	HU37	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.79			
C5	HU20	1.00	1.00	1.00	1.00	1.0 0	1.00	1.00	1.00	1.00	1.00	1.00	1.00
C6	HU17	1.00	1.00	1.00	1.00	0. 03							
	HU36					0.97	1.00	1.00	1.00	1. 00	1.00	1.00	1.00
C7	HU6	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.73		
	HU25										0.27	1.00	1.00
C1_HL	HU30	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00

9.4.2 Discussion

The value of the demand violation in this case study is much better (less) than in the operational schedule. The value of demand violation in the “Base” scenario is only 1% of the objective value without penalties. The imposition of higher penalties in the “High_Pen” scenario does improve (decrease) the value of demand violation. However, given that the excess is already low, this decrease is not large in absolute terms.

The “Force_Crews” scenario gives an objective value only 5% worse (higher) than the “Base” scenario. The “Force_Crews” solution is much more reasonable from an operational perspective, because the crew movements are reduced. NZC may want to look at the true costs of moving the crews to decide which of the scenarios to implement.

9.5 Australian Company (AC)

Australian Company (AC) is a large forestry company based in Australia. This case study models a part of the forests under their control. AC were interested in this case study for several reasons.

- In the studied region there are 10 different customers and a wide variety of log-types.
- There are two species of trees in the region and a variety of terrain.
- There are production-thinning operations to schedule as well as clearfelling.
- The large number of customers, log-types and pricing points can make it difficult to compare the effects of any production decision.
- The forest estate produces a high percentage of large diameter logs. Unfortunately, their customers cannot use all of these logs. Often the larger logs are left in log-stocks until they are downgraded and sold for a lower price.

These factors make the problem quite difficult to solve manually and lead to a marked improvement when the solution is found using the optimisation algorithm presented in this thesis. Different pricing points and methods of payment, make it difficult for AC to identify good solutions quantitatively.

The period for this case study was October 2000. The case study included:

- 7 crews;

- 26+10 harvest units (ten harvest units are early thinning);
- 81 log-types;
- 10 different customers;
- clear fell and thinning operations;
- one species.

Only one species was modelled, as the second species comprised only a small portion of the harvest.

The early (first and second) thinning operations were modelled by removing the crew C7 in the third period (as it was assigned to thinning) and removing the harvest units with thinning operations

9.5.1 Data collection

A team of three scientists from *forest research* spent a week collecting data from AC, by working closely with various people within the company. They collected the parameter data for the formulation described in Chapter 6. The parameter data were compiled in several Excel workbooks and inventory assessment information was compiled in a MARVL database.

Information on other factors that influence the decisions of the harvest scheduler were collected, for example, contractual obligations. Some of these factors were included in the OHS model as new constraints. These added constraints included complications to the crew cost calculations and constraints on log-type SED.

9.5.1.1 Period details

During the period only 30, out of 81, log-types were supplied to 10 customers, unused products were removed. One crew (C7) spent the third week harvesting first thinning stands. It was removed from the problem in that week.

AC had previously accumulated harvested logs at the skid sites in log-stocks. A decision had been made to reduce the amount of log-stocks. Therefore, the crews stopped harvesting in the fourth week.

The initial harvest unit for each crew was the harvest unit where the crew was harvesting before the initial week.

9.5.1.2 Data validation and analysis

Once the data were obtained from AC, the specific data needed for the HSA were calculated, including:

- at-roadside prices for logs in every stand - these prices were used to generate initial yields of the harvest areas;
- crew harvest unit productivity and compatibility matrices - taking into account the different harvest unit terrain and crew capabilities;
- market demands - these were altered to give some flexibility in the solution. Demands were not strictly limited to the number of truckloads given in the parameter data;
- the harvest unit area - this information was updated to consider the area harvested prior to the case study. MARVL assessment data gave the total area of the harvest areas.

9.5.1.3 Cutting strategy selection

The cutting strategies were created using the random stratified method (personal communication Murphy, 2001) mentioned in Section 8.5.3.

Seven random strategies were created for each harvest unit. Initial yields were generated using roadside prices for each harvest unit. Subsequent yield generation used the price change strategy described in Section 8.5.4.2.

9.5.1.4 Yield iterations

It was impossible to find a solution that met the market demand constraints using only yield predictions generated using roadside prices. This was also the case with solutions for the NZC operational case study (see Section 9.3). Once yields were recalculated several times, a near optimal solution to the RLP was found.

9.5.1.5 Downgrading

This case study included a large volume of high-value log-stocks, thus the ability to downgrade was an essential part of the solution method. Without this ability, all of the scenarios would have large excess volumes for logs in log-stocks that were not sold during the time horizon.

Downgrades were found for most of the log-types. These downgrades allowed logs to be interchanged between different log-types. This contrasts with the NZC operational case study, where volume was only downgraded to pulp logs. All scenarios for this case study allow the initial log-stocks to be downgraded. The two scenarios “Manual_DG” and “DG” also allow log volume harvested in the time horizon to be downgraded. The bound gap used for these scenarios was 5%, except for the “Strategy” scenario where a 10% bound gap was used so that a solution was found in reasonable time. When the “Strategy” scenario was solved with a 5% bound gap the solution had not been found after an overnight run.

9.5.1.6 Additional periods

To reduce the influence of end-effects (see Section 8.6) in the four periods in the study an extra four periods were added to the end of the time horizon. The demand data in these four periods was a duplicate of the original periods’ data. The crew allocation tables below, however, do not show the solution for the extra four periods.

The NZC operational case study only included one extra week. However, four extra weeks were needed in the AC case study because of the atypical treatment of log-stocks in this case-study. The log-stocks are drastically reduced in the fourth period (because there is no crew production), so the extra periods are needed to ensure the solution is still reasonable.

9.5.2 Results

Table 9.10: AC results for different scenarios

	Manual	Manual_Strat	Manual_DG	Unrestricted	DG	Base	Strategy
RLP Objective	-4,113	-4,113	1,399,308	3,414,156	1,771,504	1,598,726	1,598,726
Objective	-4,113	-83,376	1,346,491	3,400,996	1,754,963	1,554,879	1,385,401
Objective without penalty ($)	1,823,237	1,835,176	2,078,422	3,400,996	2,078,718	1,906,526	1,882,402
Value of demand violation ($)	826,953	851,510	293,490	4,020,109	81,907	110,277	188,708
Bound gap (%)	0	1,927.1	3.77	0.39	0.93	2.74	13.34
Solve time (secs)	11.77	1,187.73	20.73	9.11	75.14	154.2	1,206.38

To find the gains of implementing the OHSA, several different scenarios are contrasted in

Table 9.10

- Manual: The crews remain in the initial harvest units. Maximum and minimum demand restrictions are used. No yield generation is used, but the crew can use several cutting strategies in a period. Downgrading of the initial log-stocks is allowed.
- Manual_Strategy: The "Manual" scenario, with only a single cutting strategy used per period.
- Manual_DG: "Manual" scenario, with the ability to downgrade all the available volume.
- Unrestricted: An optimised scenario where there are no demand restrictions. Yield generation is allowed and log-stocks may be downgraded.
- DG: The "Base" scenario with downgrades allowed for all available volume.
- Base: The "Unrestricted" scenario with the addition of maximum and minimum demands.
- Strategy: The "Base" scenario with only compatible cutting strategies used (see Section 6.4.3) by a crew in a single period.

The bound gap for all scenarios except "Manual_Strategy" is reasonably small. The extremely large value for the bound gap in the "Manual_Strategy" scenario is caused by the very small absolute value of the RLP objective ($-4,113). The objective value is so small because the penalties almost exactly cancelled out the revenue. A percentage calculation of the bound gap, therefore, gives very misleading results, when contrasted with scenarios with larger absolute values of the objective.

In Table 9.11, the solutions for four periods of the "Manual" scenario are shown. Note, that the crews do not move from their initial harvest units, but do use a number of cutting strategies.

In Table 9.12, two crews have shifted from their original harvest units. C5 moves to a different harvest unit at the beginning of the period, and C4 moves into HU14.

Table 9.13 shows the effect of restricting the cutting strategies used in a period. In the "Strategy" scenario, only compatible yield predictions (different iterations of the same cutting strategy) may be used by a crew in a period.

9.5.3 Discussion

The difference between the manual scenarios and the optimised scenarios in this case study is most apparent when comparing the value of the demand violations. The combination of yield generation and crew allocation in the optimised scenarios improves (lessens) the value

Table 9.11: AC crew allocations for the "Manual" scenario

Crew	Harvest Unit	Strategy	1	2	3	4
C1	HU1	HU1-2	0.52	0.38	0.18	
		HU1-7	0.48	0.62	0.82	
C2	HU8	HU8-2		0.20	0.20	
		HU8-3			0.18	
		HU8-4			0.21	
		HU8-7	1.00	0.80	0.41	
C3	HU23	HU23-2	0.23		0.00	
		HU23-3	0.43	0.66	0.01	
		HU23-4	0.34	0.34	0.34	
		HU23-6			0.66	
C4	HU11	HU11-2			0.02	
		HU11-5			0.07	
		HU11-6	0.28	0.20	0.61	
		HU11-7	0.72	0.80	0.31	
C5	HU22	HU22-4	1.00	1.00	1.00	
C6	HU27	HU27-3	1.00	0.99	1.00	
		HU27-4		0.01		
C7	HU29	HU29-3	0.32	0.97		
		HU29-6	0.68	0.03		

Table 9.12: AC crew allocations for the "Base" scenario

Crew	Harvest Unit	Strategy	Iter	1	2	3	4
C1	HU1	HU1-1	1	0.0 1	0.04	0.18	
			2	0.23	0.24	0.61	
		HU1-2	1	0.07			
			2		0.07	0.21	
		HU1-3	3	0.69	0.61		
		HU1-7	2		0.05		
C2	HU8	HU8-5	2		0 .13	0.13	
			3	0.09	0.12	0.12	
		HU8-6	2	0.24	0.13		
			3	0.68	0.21	0.21	
		HU8-7	3		0.42	0.55	
C3	HU23	HU23-2	1	0.45	0.02	0.02	
			2	0.00	0.22	0.41	
		HU23-4	2		0.14	0.27	
		HU23-5	2	0.08	0.26	0.30	
		HU23-6	1	0.47	0.36		
C4	HU11	HU11-6	2	0.74			
		HU11-7	1	0.26			
	HU14	HU14-4	3		1.00	0.67	
		HU14-6	1			0.33	
C5	HU4	HU4-1	2		0.23		
		HU4-3	1	1.00	0.77	0.78	
		HU4-4	2			0.22	
C6	HU27	HU27-4	1		0.13		
		HU27-5	3	1.00	0.87	1.00	
C7	HU29	HU29-6	2	1.00	1.00		

of demand violation noticeably. The value of demand violation in the "Base" scenario is 86% better than the "Manual" solution.

Table 9.13: AC Crew Allocations for the "Strategy" Scenario

Crew	Harvest Unit	Strategy	Iter	1	2	3	4
C1	HU1	HU1-1	1			0.18	
			2			0.82	
		HU1-2	1		0.02		
			2		0.98		
		HU1-3	1	0.31			
			3	0.69			
C2	HU8	HU8-6	2			1.00	
		HU8-7	2		0.15		
			3	1.00	0.85		
C3	HU23	HU23-4	2		1.00	1. 00	
		HU23-6	1	0.80			
			3	0.20			
C4	HU14	HU14-1	2			1.00	
		HU14-4	1	0.08	0.09		
			3	0.92	0.91		
C5	HU4	HU4-1	3			1.00	
		HU4-3	1	1.00	1.00		
C6	HU27	HU27-5	2		1.00	1.00	
			3	1.00			
C7	HU29	HU29-6	2	1.00	1.00		

Table 9.14 compares the effects of yield generation, downgrading and crew allocation. The starred (*) entries have been generated for this table only, while the other entries are extracted from Table 9.10. This table shows that the effect of crew allocation gives an improvement in demand violation, that varies between a 23-45%. The improvement from downgrading is between 65 and 74%, while yield generation gives an improvment of between 76 and 83%. These methods do not combine linearly, however, and their combined effect is between 84 and 87%. These two methods have very similar effects on the solution and therefore were not expected to combine together linearly.

Table 9.14: AC Comparison of demand violation with crew allocation, yield generation and downgrading

	Crew Allocations	
	Manual	Optimised
No downgrade or yield generation	826,953	(*)634,360
Downgrade only	293,490	(*)162,415
Yield Generation Only	(*)196,851	110,277
Downgrade and yield generation	(*)132,840	81,907

Even though downgrading combined with yield generation does improve the solution, the argument advanced in Section 3.5.4 that downgrading should be limited to initial log-stocks (see Section 3.5.4) is still valid. Yield generation more accurately models the harvesting process, lower value logs will be bucked from the stem when they are needed. While downgrading assumes that the logs will be bucked as high value log-types and then sold as lower value log-types.

Crew-strategy branches generate solutions that will be acceptable to harvesting crews (see Section 6.4.3). The two scenarios where Crew-strategy branches are used are “Strategy” and “Manual_Strat”. Restricting the cutting strategies in this way does lead to worse objective values. The “Strategy” scenario shows that the objective value without penalties (reduced by 1.3%) and the demand violation (increases by 72%) is worse when compared to the “Base” scenario. The increase in demand violation is expected purely because the ability to produce the correct mix of log-types is reduced. Importantly however, the value of the violation in the “Strategy” scenario is still less than any of the manual solutions. The “Strategy” scenario has a 78% improvement in demand violation over the “Manual_Strat” scenario. These demand violations are compared in Table 9.15 with data extracted from Table 9.10.

Table 9.15: AC Comparison of demand violation with crew allocation and strategy branches

	Crew Allocations	
	Manual	Optimised
No strategy branches	826,953	110,277
Strategy branches	851,510	188,708

The optimisation improves (increases) the objective value without penalties in this case study. The “Base” scenario is 5% better than the “Manual” scenario. When cutting strategies are restricted, the “Strategy” scenario is 3% better than the “Manual_Strat” scenario.

9.6 Conclusion

The case studies for NZC and AC show that the OHSA is capable of providing solutions to realistic problems. The NZC operational schedule demonstrated the speed and efficiency of the algorithm on a very small problem. The NZC annual schedule showed that the OHSA could be used to solve problems with a longer time horizon than it was initially intended.

The AC case study shows that the OHSA gives good results for moderately sized problems.

The use of yield generation was also proven effective. By solving both the NZC and AC problems, the OHSA was shown to be flexible enough to be implemented in two different forestry companies.

Chapter 10

Discussion

And 'mid these dancing rocks at once and ever
It flung up momently the sacred river.
(Coleridge 1798)

This thesis has presented significant research that applies optimisation techniques to the field of Forestry Operational Harvest Scheduling.

This chapter will:

- briefly outline the OHS problem that has been considered;
- highlight the contributions developed in this thesis;
- demonstrate through case studies that the solution methods solve practical problems;
- discuss some directions for future research.

10.1 Problem description

An OHS:

- assigns forest harvesting crews to locations within a forest in the short-term (4-8 weeks);
- instructs crews to harvest specific log-types and allocates these log-types to customers;
- maximises profitability while meeting customer demand.

A full problem description is given in Chapter 3. In this thesis we solve a problem which includes:

- a forest wide scope;
- unambiguous crew locations;
- multiple periods and crew movements;
- the ability to move crews mid-period;
- iterative generation of possible crew production;
- the distribution of logs to customers;

The problem includes all the important aspects that have been included in previous literature (see Chapter 4) and adds new features discussed later in Section 10.2.

10.2 Contributions

This thesis makes contributions in two areas, the modelling of OHS problems and the solution strategies applied.

10.2.1 Modelling

The problem formulation, in this thesis, is given in Chapter 6. The use of a Model II formulation (see Section 6.2.2) for the CA sub-model is unique within the OHS literature. Because of this novel formulation, methods that have previously been applied to GSPP formulations of rostering problems, may be applied to the OHS problem. These methods include column generation and constraint branching.

The Model II formulation follows the crews through time. Therefore, crew movements are easy to model within a crew schedule's structure. The implicit constraints discussed in Section 7.2 including those that force a crew to remain in a harvest unit until completion are also easily modelled. The ability to allow crews to move mid-period described in Section 7.1.1 can only be implemented in a Model II formulation.

The PT sub-model presented in this thesis is similar to other models given in the literature, for instance, Murphy (1998) and Epstein et al. (1999b). It includes all of the common features found in the literature as well as extensions that model SED and product fraction restrictions. However, the PT sub-model will be altered within a particular forestry company, as the company will have different operating practices.

10.2.2 Computational techniques

To solve the formulation described in Chapter 6 the techniques described in Chapter 7 and Chapter 8 are applied. These techniques include several new methods for solving these types of problem. The techniques include:

- relaxed integer solutions;
- yield prediction generation;
- column generation;
- constraint branching;
- integer allocation.

10.2.2.1 Relaxed integer solutions

The use of relaxed integer solutions (see Section 7.1.2), to give unrestricted movement solutions (see Section 7.1.1) is an important part of this thesis. By relaxing the definition of an integer solution, solutions can be found that allow the LP to continue to optimise important elements of the problem. The careful definition of the type of unrestricted movement solutions, allows logical consistency and unambiguous interpretation of the solution.

This technique allows:

- crews to move mid-period;
- a number of yield prediction iterations to be used in a single period when complementary crew-cutting strategy branches are applied;
- multiple cutting strategies to be used, when crew cutting strategy branches are not applied.

A relaxed integer solution that can be interpreted as an unrestricted movement solution, is only possible when a Model II formulation is used in conjunction with the constraint branches described in Section 8.3.

10.2.2.2 Yield prediction generation

The bucking optimisation literature (see Section 4.2.1) discusses techniques that meet customer demands by generating new yield predictions. In my research, these techniques were essential to reduce violation of the demand constraints. The technique used is similar to that described in

Cossens (1996) but is applied to a full OHS model similar in scope to the OPTICORT model (Epstein et al. 1999b).

The case study results in Chapter 9 suggest that the yield predictions may be altered in two ways. New yield predictions can be generated by an iterative process, or downgrading can allow volume to move between different log-types. The case study results suggest the iterative process has a marginal advantage over downgrading, but either is very effective.

From a modelling perspective the iterative process is preferable because it models the correct behaviour of the crews. In the iterative process the all logs are bucked to their final log-type specifications(see Section 2.5.1.4). With downgrading the model assumes that a log is bucked to one log-type specification and then it is transformed into another log-type before it is transported to the customer. Iterative generation also ensures production will meet the specifications of the customer. In addition, only iterative generation can strictly control the number of log-types produced by a harvesting crew.

However, the limited use of downgrading (initial log-stocks only) was necessary in the AC case study. This case study was atypical, however, because of the large amount of log-stocks.

The case studies recommend an iterative process in preference to pre-generation of yield predictions. Pre-generation of yields, naively applied, can have very little effect on demand violation. The NZC operational plan case study shows this effectively. A solution that used 16 pre-generated cutting strategies still had very high demand violations. The application of downgrading however reduced the violation by half.

10.2.2.3 Column generation

The number of possible crew schedules in a Model II formulation is very large. Column generation is necessary if realistic problems are to be solved. The dynamic creation of crew schedules has not been previously used in this field.

The column generation algorithm, described in Section 8.2, finds candidate entering columns. It uses a shortest path formulation to find crew schedules with negative reduced costs. Pre-generation of some of the values used in the algorithm results in significant reduction in solution time. Properties of the reduced cost calculation can reduce the number of predecessors considered at each stage and therefore reduce the time taken to generate new crew schedules. Implementation of the integer feasible column restriction (see Section 8.2.4), also improves the performance of the B&B.

10.2.2.4 Constraint branching

The constraint branching strategies described in Section 8.3 were used to find a solution. Constraint branches were constructed to give relaxed integer solutions, and to allow column generation to continue within the B&B process. The repeated application of a typical constraint branch will force the integer relaxation of a problem to become integer feasible or infeasible. The constraint branches in this thesis, however, force the relaxation to become *relaxed integer* feasible (see Section 7.1.2) or infeasible.

10.2.2.5 Integer Allocation

The integer allocation algorithm described in Section 8.4 is very effective. In most cases, integer solutions are found by the integer allocation, before the B&B tree has found an integer node. The integer allocation is especially effective when it finds a good solution at the root node of the B&B tree. The objective value of this solution can be used to bound a large proportion of the B&B tree.

In contrast to other applications of integer allocation heuristics, the algorithm in this thesis uses repeated calls to the simplex algorithm and the branching strategies used in the main B&B algorithm. This type of approach does take a longer time to find an allocated solution, but it is necessary because of the use of LP to find the values of decisions in the PT sub-model. Since these non-optimal uses of the branch and bound techniques occur in parallel to the main B&B algorithm the overall solution process retains the ability to find optimal solutions (if the bound gap is set to 0).

10.3 Case studies

The results from the three case studies show that the Operational Harvest Scheduling Algorithm (OHSA) can be applied to realistic problems. All three case studies were derived from forestry companies. They represent the actual OHS problem that was solved in the company, in the time period given.

The NZC operational schedule (see Section 9.3), showed that the OHSA algorithm could quickly find the solutions to small problems. The optimisation ("Demand" scenario) is able to meet the demand constraints better than the manual ("Manual_DG" scenario) solution. The optimisation reduces the value of demand violation by 15% compared to the manual solution. The impact of downgrading in this case study is especially evident, as there is no iterative yield generation. A solution without downgrading increases demand violation by 75%.

The NZC annual schedule (see Section 9.4), showed the operation of the OHSA on a problem with a time scale longer than the 1 week to 3 month time scale typical of most OHSs. Unfortunately, there was no manual solution to compare to the optimised ("Base" scenario) results. However, the case study problem was solved in reasonable time (77.95 secs). The OHSA gave good solutions with very low bound gaps (0.09 %).

The AC case study (see Section 9.5), was the most complex of the three case studies. The detailed information gathered on AC operations resulted in extensive customisation of the PT sub-model. The large initial log-stocks made some downgrading essential for all scenarios. Again, the OHSA performs well on all the scenarios. It solves the "Base" scenario in a reasonable time (154.2 secs) with small bound gaps (2.74%).

- With no cutting strategy restrictions: the optimisation ("Base" scenario) reduced the value of demand violation by 86% and improved the objective value without penalties by 5% over the manual ("Manual" scenario) solution.
- With cutting strategy restrictions: the optimisation ("Strategy" scenario) reduced the value of demand violation by 78% and improved the objective value without penalties by 3% over the manual ("Manual_Strat" scenario) solution.

These figures show the benefits of optimisation, especially its effect on demand violation. They are summarised in Table 10.1

10.4 Further work

As a result of the research presented in this thesis there are several issues that could be investigated in further research. These include topics relating to:

- yield prediction generation;

Table 10.1: Comparison of case study results

	NZC operational	NZC annual	AC no strategy restrictions	AC strategy restrictions
Objective value without penalties improvement (%)	7	n/a	5	3
Value of demand violation improvement (%)	15	n/a	86	78
Bound gap (%)	3.56	0.09	2.74	13.34
Solution time (secs)	8.47	77.95	154.2	1,206.38

- uncertainty in yield predictions;
- the use of a rolling horizon OHS model;
- an investigation of the merits of priority list yield generation and dynamic programming yield generation.

In this thesis, the yield predictions were generated by MARVL, a software product provided by *forest research* (see Section 8.5). Because the algorithms within this software were not under my direct control, the iterative yield generation was not implemented as efficiently as it might have been. In particular, the yield generation step was not placed within the column generation framework (see Section 8.2). It would be interesting to see the effect of placing yield generation algorithms directly within the column generation.

The properties of a yield prediction frontier are described in Section 3.3. It may be possible to represent the yield prediction frontier for the harvest unit *before* an OHS problem is constructed. The ability to use a yield prediction frontier with the column generation step would significantly improve the performance of the yield generation iteration.

Section 3.3.2 discusses some concerns about uncertainty of yield predictions. This subject is also discussed in Section 4.3. How to place uncertainty into OHS problems is certainly an open question. Investigation of techniques such as robust optimisation or perhaps stochastic optimisation would be a basis for good future work

The OHS problem is used as a rolling horizon model (see Section 8.6). In this thesis only one time horizon for each scenario is ever solved, the periods are never ‘rolled-over’. A detailed examination of the implementation issues found by using the model in this way could be completed. This examination would look at how the solution of the model changes as it is ‘rolled-over’ and what kinds of constraints are needed to keep the model consistent over several weeks. The effect of end-effects on the model could also be quantified.

Section 4.3 describes the division within the literature over the use of priority list generated yield predictions and dynamic programing generated yield predictions. This thesis only deals with dynamic programing generated yield predictions. These are more useful as they can generate a yield prediction frontier because of their proof of optimality (see Section 3.3.1). An investigation of the accuracy of both methods by comparing them to actual production would provide an excellent basis to evaluate their differences.

Appendix A

Large Format Matrix Layout

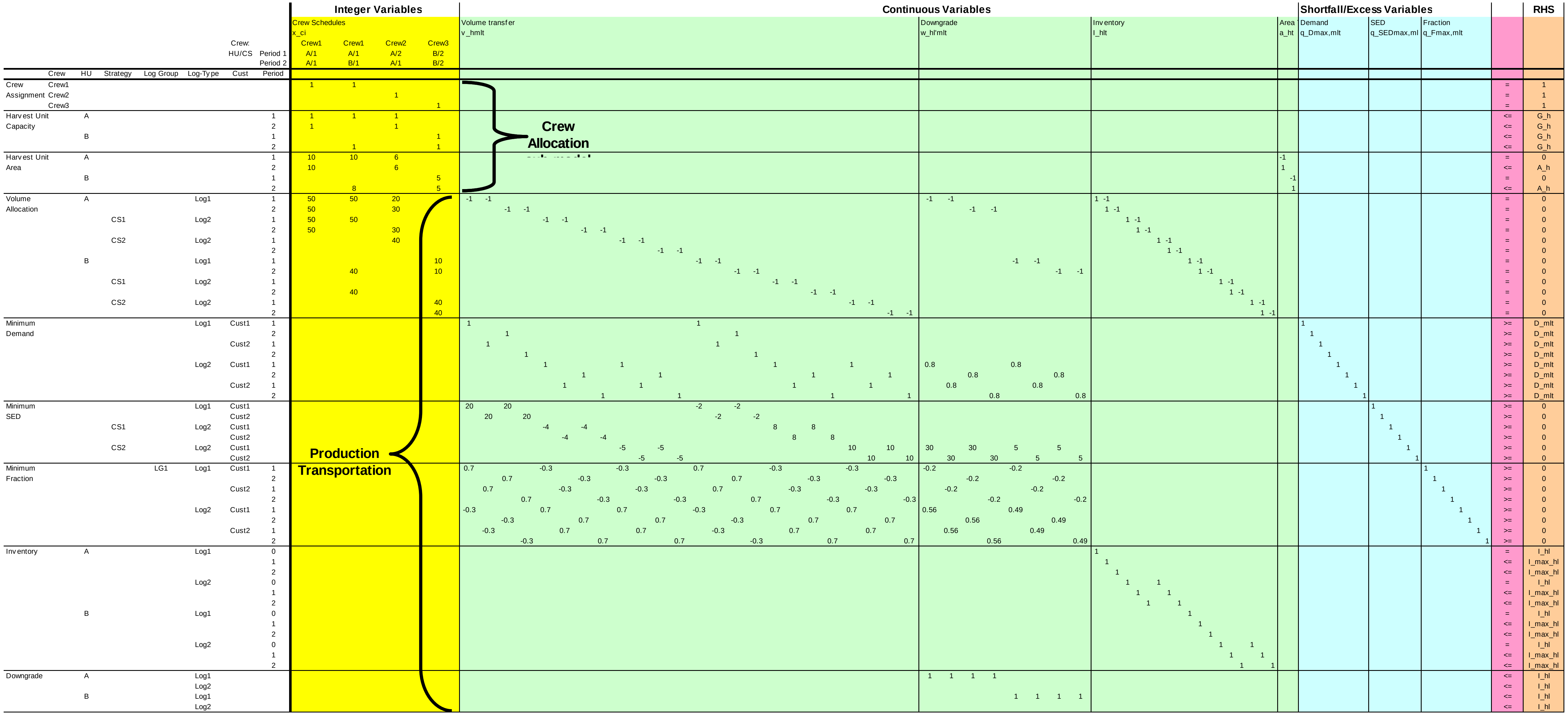


Figure A.1: Large format matrix layout, excluding maximum demand, fraction and SED constraints

Appendix B

Derivation of Uncertainty in Fractional Harvest Unit Yields

The confidence intervals supplied from a inventory system like MARVL(Deadman & Goulding 1979) are calculated from the standard deviation of volume per hectare of the entire stand (s_s). This is equal to the standard deviation of the volume estimate for the stand (S_s) divided by the area of the stand (a_s).

$$s_s = \frac{S_s}{a_s}$$

In the OHS only a fraction of the stand is harvested in a period. To calculate the precision of the per hectare estimate for this smaller fraction of the stand the standard deviation of smaller area (s_f) is found.

If we model the variation from two sources, the between plot variation and the between tree variation.

$$s_s = \sqrt{\frac{s_p^2}{n_{plots}} + \frac{s_t^2}{n_{trees}}} \tag{B.1}$$

where:

s_s standard deviation of the stand volume per hectare estimate;
s_p^2 variance between plots;
s_t^2 variance between trees;
n_{plots} number of trees in the stand;
n_{trees} number of plots in the stand.

If we assume that the number of plots (n_{plots}) is directly related to area and the same applies to the number of trees (n_{trees}).

$$\frac{n_{plots}}{nf_{plots}} = \frac{n_{trees}}{nf_{trees}} = \frac{a_s}{a_f}$$

where:

nf_{plots} number of trees in the fraction;
nf_{trees} number of plots in the fraction;
a_s area of the stand;
a_f area of the fraction.

If we also assume that the plot properties are not spatially correlated, i.e., plots close to each other are not similar and the plot and tree variations are independent [1].

$$s_f = \sqrt{\frac{s_p^2}{n_{plots}}\frac{a_s}{a_f} + \frac{s_t^2}{n_{trees}}\frac{a_s}{a_f}}$$

therefore,

$$s_f = s_s\sqrt{\frac{a_s}{a_f}} \tag{B.2}$$

where:

s_f is the standard deviation of the fraction per hectare.

Note, Equation (B.2) also holds true if only plot variation is modelled, which is common in inventory systems.

In order to check this result we can derive the variation of the per hectare stand estimates (s_s). If the standard deviation of the entire stand is $S_s = s_s a_s$, and the total standard deviation of the fraction is $S_f = s_f a_f$.

$$S_s^2 = \sum S_f^2$$

[1]Note, these simplifying assumptions are used because no spatial data are available.

If we assume that all the fractional variations are equal, because we have no other data available

$$S_s^2 = \frac{a_s}{a_f} S_f^2.$$

Therefore,

$$S_s = \sqrt{\frac{a_s}{a_f}} S_f$$

and

$$s_s a_s = \sqrt{\frac{a_s}{a_f}} s_f a_f$$

and

$$s_s = \frac{a_f}{a_s} \sqrt{\frac{a_s}{a_f}} s_f$$

or

$$s_s = \sqrt{\frac{a_f}{a_s}} s_f.$$

Therefore, Equation (B.2) is correct.

Appendix C

Use of Lattice Path Problems to Calculate Complexity of Shortest Path Formulations

If the harvested area of each harvest unit is considered as a state in the Column generation subproblem (Section 8.2). This can be approximated by counting the number of equal length periods a crew has visited each harvest unit. We, however, do not need to know the order that the crew visited these harvest units.

If a vector is defined that counts the number of times a harvest unit has been visited. The number of possible vectors for each harvest unit in period t can be described in terms of the lattice path problem. This problem can be defined as the number of individual paths from the origin of a cartesian lattice to the point $(n-1, t)$ of length $(n-1)+t$ only moving horizontally or vertically from point to point. The number of different vectors at stage t in a problem with n harvest units is $\binom{(n-1)+t}{n-1}$.

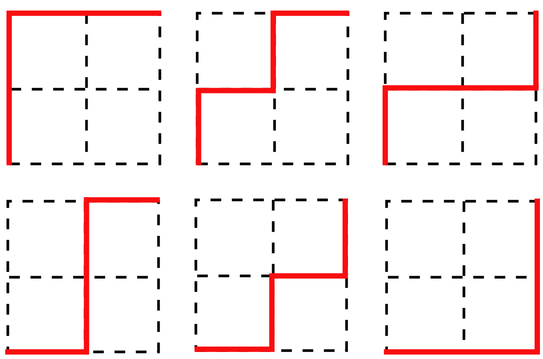


Figure C.1: Diagram of the lattice paths in a 2x2 lattice

The $(n-1)$ term comes from the difference between the 0-based counting of periods in the lattice and the 1-based counting of the harvest units. For example, the 2x2 lattice in Figure C.1 models a problem with three harvest units on the horizontal axis and two periods in the vertical axis.

This will mean that the total number of states (this includes the previous harvest unit and the vector of visited harvest units) in the dynamic program at stage t is given by

$$States = n\binom{(n-1)+t}{n-1}$$

For a reasonable size problem with 36 harvest units and 8 periods the number of states will be 5,220,306,468 which is much to large to generate columns in a reasonable time.

This number is reduced if we take into account that the harvest units will not be available for harvesting for the entire 8 periods as the harvest unit may be completed in only four periods.

Appendix D

Kubla Khan

Kubla Khan
Or, a Vision in a Dream. A Fragment.

In Xanadu did Kubla Khan
A stately pleasure-dome decree:
Where Alph, the sacred river, ran
Through caverns measureless to man
Down to a sunless sea.
So twice five miles of fertile ground
With walls and towers were girdled round:
And there were gardens bright with sinuous rills,
Where blossomed many an incense-bearing tree;
And here were forests ancient as the hills,
Enfolding sunny spots of greenery.

But oh! that deep romantic chasm which slanted
Down the green hill athwart a cedarn cover!
A savage place! as holy and enchanted
As e'er beneath a waning moon was haunted
By woman wailing for her demon-lover!
And from this chasm, with ceaseless turmoil seething,
As if this earth in fast thick pants were breathing,

A mighty fountain momently was forced:
Amid whose swift half-intermitted burst
Huge fragments vaulted like rebounding hail,
Or chaffy grain beneath the thresher's flail:
And 'mid these dancing rocks at once and ever
It flung up momently the sacred river.
Five miles meandering with a mazy motion
Through wood and dale the sacred river ran,
Then reached the caverns measureless to man,
And sank in tumult to a lifeless ocean:
And 'mid this tumult Kubla heard from far
Ancestral voices prophesying war!
The shadow of the dome of pleasure
Floated midway on the waves;
Where was heard the mingled measure
From the fountain and the caves.
It was a miracle of rare device,
A sunny pleasure-dome with caves of ice!
A damsel with a dulcimer
In a vision once I saw:
It was an Abyssinian maid,
And on her dulcimer she played,
Singing of Mount Abora.
Could I revive within me
Her symphony and song,
To such a deep delight 'twould win me,
That with music loud and long,
I would build that dome in air,
That sunny dome! those caves of ice!
And all who heard should see them there,
And all should cry, Beware! Beware!
His flashing eyes, his floating hair!
Weave a circle round him thrice,
And close your eyes with holy dread,

For he on honey-dew hath fed,
And drunk the milk of Paradise.

Samuel Taylor Coleridge (1798)

List of References

Barros, O. & Weintraub, A. (1982), 'Planning for a vertically integrated forest industry', *Operations Research* **30**, 1168–1182.

Bazzaraa, M. S., Jarvis, J. J. & Sherali, H. D. (1990), *Linear Programming and Network Flows*, John Wiley & Sons, New York.

Bettinger, P., Boston, K. & Sessions, J. (1999), 'Intensifying a heuristic forest harvest scheduling search procedure with 2-opt decision choices', *Canadian Journal of Forest Research* **29**, 1784–1792.

Bevers, M. J., Hof, J., Uresk, D. W. & Schenback, G. L. (1997), 'Spatial optimisation of prairie dog colonies for black-footed ferret recovery', *Operations Research* **45**(4), 495–507.

Borges, J. G., Hoganson, H. M. & Rose, D. W. (1999), 'Combining a decomposition strategy with dynamic programming to solve spatially constrained forest management scheduling problems', *Forest Science* **45**(1), 201–212.

Boston, K. & Bettinger, P. (1999), 'An analysis of Monte Carlo integer programming, simulated annealing, and tabu search heuristics for solving spatial harvest scheduling problems', *Forest Science* **45**(2), 292–301.

Boston, K. & Kiser, J. (1999), Just-in-time operational scheduling system, *in* 'Conference Proceedings Timber Harvesting and Transportation Technologies for Forestry in the New Millenium', Pietermaritzburg, South Africa, pp. 24–34.

Brooke, A., Kendrick, D. & Meeraus, A. A. (1992), *GAMS–A User's Guide (Release 2.25)*, The Scientific Press, San Francisco, CA.

Buford, M. A., ed. (1991), *Proceedings of the 1991 Symposium on Systems Analysis in Forest Resources*, Southeastern Forest Experiment Station, Asheville NC.

Clocksin, W. F. & Mellish, C. S. (1994), *Programming in PROLOG*, 4 edn, Springer Verlag.

Coleridge, S. T. (1798), 'Kubla khan'. Or, a Vision in a Dream. A Fragment.

Cossens, G. P. (1996), Optimisation of short term log allocation, Master of applied science thesis, Lincoln University.

Cossens, P. (1992), Planning and control of short term log allocation in New Zealand, *in* 'Integrated Decision Making in Planning and Control of Forest Operations, Proceedings of IFURO Conference', School of Forestry, University of Canterbury, Christchurch, New Zealand, pp. 46–56.

CPLEX (1994), *CPLEX Users Manual*, CPLEX Optimisation Inc, Incline Village, NV.

Dantzig, G. B. (1963), *Linear Programming and Extensions*, Princeton University Press, Princeton, NJ.

Dantzig, G. B. & Wolfe, P. (1960), 'Decomposition principles for linear programs', *Operations Research* **8**.

Davis, L. & Barrett, R. H. (1993), Spatial integration of wildlife habitat analysis with long-term forest planning over multiple-owner landscapes, *in* D. C. LeMaster & R. A. Sedjo, eds, 'Modeling Sustainable Forest Ecosystems', American Forests, Washington, DC, pp. 143–154.

Deadman, M. W. & Goulding, C. J. (1979), 'A method for the assessment of recoverable volume by log types', *New Zealand Journal of Forest Science* **9**, 225–239.

Dilworth, J. R. & Bell, J. F. (1984), *Log Scaling and Timber Cruising*, OSU Bookstores Inc., Corvallis, OR.

du Merle, O., Villeneuve, D., Desrosiers, J. & Hansen, P. (1999), 'Stabilized column generation', *Discrete Mathematics* **194**, 229–237.

Eng, G., Daellenbach, H. & Whyte, A. G. D. (1986), 'Bucking tree-length stems optimally', *Canadian Journal of Forest Research* **16**, 1030–1035.

Epstein, R., Morales, R., Seron, J. & Weintraub, A. (1999a), 'Use of OR systems in the Chilean forest industries', *Interfaces* **29**(1), 7–29.

Epstein, R., Nieto, E., Weintraub, A., Gabarro, J. & Chevalier, P. (1999b), ‘A system for the design of a short term harvesting strategy’, *European Journal of Operational Research* **119**, 427–439.

Fisher, G. W. (1995), Sampling methods for inventory, *in* Hammond (1995), pp. 108–109.

Fourer, R., Gay, D. M. & Kernighan, B. W. (1993), *AMPL: A Modeling Language for Mathematical Programming*, Duxbury Press / Brooks/Cole Publishing Company.

García, O. (1984), FOLPI, a forestry-oriented linear programming interpreter, *in* H. Nagumo et al., eds, ‘Proceedings IUFRO Symposium on Forest Management Planning and Managerial Economics’, University of Tokyo, Tokyo, pp. 293–305.

García, O. (1990), ‘Linear programming and related approaches in forest planning’, *New Zealand Journal of Forest Science* **20**(3), 307–331.

Gilmore, P. C. & Gomery, R. E. (1965), ‘Multistage cutting stock problems of two and more dimensions’, *Operations Research* **13**, 94–120.

Glover, F., Kelly, J. P. & Laguana, M. (1995), ‘Genetic algorithm and tabu search: Hybrids for optimisation’, *Computer Operations Research* **22**, 111–134.

Glover, F. & Laguna, M. (1993), Tabu search, *in* C. R. Reeves, ed., ‘Modern Heuristic Techniques for Combinatorial Problems’, Blackwell Scientific Publications, Oxford, pp. 70–150.

Goulding, C. J. (1974), Linear programming a multicommodity distrubution system for forest produce., Forest Mensuration Report (unpublished) 43, New Zealand Forest Service, Forest Research Institute.

Gunn, E. A. (1991), Some aspects of hierarchical production planning in forest management, *in* Buford (1991), pp. 54–62.

Hammond, D., ed. (1995), *1995 Forestry Handbook*, N.Z Institute of Forestry, Christchurch NZ.

Harmon, P., Maus, R. & Morrisey, W. (1988), *Expert Systems: Tools and Applications*, John Willey & Sons, New York.

Hipkins, G. (1995), Log specifications and the customer, *in* Hammond (1995), pp. 191–192.

IFR Logger (2002), WWW page: http://www.ifrtech.com/ifrlogger.htm.

Johnson, K. N. & Scheurman, H. L. (1977), 'Techniques for prescribing optimal timber harvest and investment under different objectives — discussion and synthesis', *Forest Science Monograph* **18**.

Johnson, K., Stuart, T. & Crimm, S. (1986), *FORPLAN Version 2: An Overview*, USDA Forest Service, Land Management Planning Systems Section, Washington D.C.

Kirby, M. W., Wong, P., Hager, W. A. & Huddleston, M. E. (1980), *A Guide to the Integrated Resources Planning Model*, ASDA Forest service, Berkeley, CA.

Laroze, A. (1999b), 'A linear programming, tabu search method for solving forest-level bucking optimisation problems', *Forest Science* **45**(1), 108–116.

Laroze, A. & Greber, B. (1991), Multi-level harvest planning and log merchandising using goal programming, *in* Buford (1991), pp. 24–30.

Laroze, A. & Greber, B. (1993), Using Monte-Carlo simulation to generate rule-based bucking patterns, *in* Paredes (1993), pp. 153–165.

Laroze, A. J. & Greber, B. (1997), 'Using tabu search to generate stand-level, rule-based bucking patterns', *Forest Science* **43**(2), 157–169.

Maclaren, J. P. (1993), *Radiata Pine Growers' Manual*, New Zealand Forest Research Institute, Rotorua, New Zealand.

M.A.F. (2002), 'New Zealand forest industry facts and figures', Ministry of Agriculture and Forestry Information Booklet.

Manley, B. R. & Threadgill, J. A. (1987), Testing FOLPI, a linear-programming-based forest estate model, *in* J. A. Kininmonth, ed., 'Proceeding of Conversion Planning Conference', N.Z. Ministry of Forestry, pp. 275–302. FRI Bulletin No. 128.

Manley, B. R. & Threadgill, J. A. (1991), 'LP used for valuation and planning of New Zealand plantation forests', *Interfaces* **21**(6), 66–79.

Martell, D., Gunn, E. & Weintraub, A. (1998), 'Forest management challenges for operational researchers', *European Journal of Operational Research* (104), 1–17.

McDill, M. E. & Braze, J. (2001), 'Using the branch and bound algorithm to solve forest planning problems with adjacency constraints', *Forest Science* **47**(3), 403–418.

McGuigan, B. N. (1984), ‘A log resource allocation model to assist forest industry managers in process selection and location, wood allocation and transportation and production planning’, *APPITA* **37**(4), 289–296.

McNaughton, A. (1998), Long-Term Scheduling of Harvesting with Adjaency and Trigger Constraints, PhD thesis, Dept of Engineering Science, University of Auckland.

Mendoza, G. A. & Bare, B. (1986), ‘A two-stage decision model for log bucking and allocation’, *Forest Products Journal* **36**(10), 70–74.

Murphy, G. (1998), ‘Allocation of stands and cutting patterns to logging crews using tabu search heuristics’, *Journal of Forest Engineering* **9**, 31–37.

Murray, A. T. (1999), ‘Spatial restrictions in harvest scheduling’, *Forest Science* **45**, 45–52.

Murray, A. T. & Church, R. L. (1995*a*), ‘Analysing cliques for imposing adjacency restrictions in forest models’, *Forest Science* **41**(3).

Murray, A. T. & Church, R. L. (1995*b*), ‘Heuristic solution approaches to operational forest planning problems’, *OR Spektrum* **17**, 193–203.

Nelson, J. & Brodie, J. D. (1990), ‘Comparison of a random search algorithm and mixed integer programming for solving area-based forest plans’, *Canadian Journal of Forestry Research* **20**, 934–942.

Nelson, J., Brodie, J. D. & Sessions, J. (1988), Integrating short-term spacially feasible harvest plans with long-term harvest schedules using monte-carlo integer programming and linear programming, General technical report, USDA Rocky Mountains Forestry Range Experimental Station. pp. 224-229.

Ogweno, D. C. O. (1995), Integrated Optimization of Operational and Tactical Planning for Log Production, Phd thesis in forestry, University of Canterbury.

O’Hara, A. J., Faaland, B. H. & Bare, B. B. (1989), ‘Spatially constrained timber harvest scheduling’, *Canadian Journal of Forest Research* **19**, 715–724.

Papps, S. R. & Manley, B. R. (1992), Integrating short-term planning with long-term forest estate modelling using FOLPI, *in* G. Wood & B. Turner, eds, ‘Integrating Forest Information Over Space and Time’, IUFRO, ANUTECH Pty Ltd, Canberra, ACT, Australia, pp. 188–198.

Paredes, V., ed. (1993), *Forest Management and Planning in a Competitive and Environmentally Conscious World: Proceedings International Symposium on System Analysis and Management Decisions in Forestry*, Universidad Austral de Chile,, Valdivia, Chile.

Parker, P., Park, R., Clement, B. & Gibbons, W. (1995), Effect of number of log grades on log making error, Report 20(23), NZ Logging Industry Research Organisation. :1-7.

Purey-Cust, J. & Hammond, D. R. (1995), A condensed history of forestry in New Zealand, *in* Hammond (1995).

Rubinstein, R. Y. (1981), *Simulation and the Monte-Carlo Method*, John Wiley & Sons, New York.

Ryan, D. (1980), Zip: A zero-one integer programming package for scheduling, Research Report CSS 85, AERE, Harwell, Oxfordshire.

Ryan, D. (1992), 'The solution of massive generalised set partitioning problems in aircrew rostering', *Journal of the Operational Research Society* **43**(5), 459–467.

Ryan, D. & Falkner, J. (1988), 'On the integer properties of scheduling set partitioning models', *European Journal of Operational Research* **35**, 442–456.

Ryan, D. M. & Osborne, M. R. (1988), 'On the solution of highly degenerate linear programmes', *Math. Programming* **41**(3, (Ser. A)), 385–392.

Scharge, L. (1993), *LINGO Optimization Modeling Language*, LINDO Systems Inc, Chicago.

Schrage, L. (1991), *LINDO Users Manual*, The Scientific Press Inc, South San Franciso, CA.

Sessions, J., Olsen, E. & Garland, J. (1989), 'Tree bucking for optimal stand value with log allocation constraints', *Forest Science* **35**(1), 271–276.

USDA Forest Service (1995), *Spectrum User's Guide*, Ecosystem Management Analysis Centre.

Van Deusen, P. C. (1999), 'Multiple solution harvest scheduling', *Silva Fennica* **33**(3), 207–216.

Weintraub, A., Barahona, F. & Epstein, R. (1994), 'A column generation algorithm for solving general forest planning problems with adacency constraints', *Forest Science* **40**(1), 142–161.

Weintraub, A. & Cholaky, A. (1991), 'A hierarchical approach to forest planning', *Forest Science* **37**, 439–460.

Weintraub, A., Cordova, F., Morales, R., Mercader, J. M., Tabensky, N. & Magendzo, A. (1993), An expert systems approach to short term forest management, *in* Paredes (1993), pp. 105–119.

Weintraub, A., Jones, G., Meacham, M., Magendzo, A., Magendzo, A. & Malchuk, D. (1995), 'Heuristic procedures for solving mixed-integer harvest scheduling - transportation planning models', *Canadian Journal of Forest Research* **25**, 1618–1626.

Weintraub, A., Morales, R., Serón, J., Epstein, R. & Traverso, P. (1991), Managing operations in pine forest industries, *in* 'Proceedings of the 1991 Symposium on Systems Analysis in Forest Resources', USDA Forest Service, Southeastern Forest Experimental Station, pp. 31–34. General Technical Report SE-74.

Weintraub, A. & Navon, D. (1976), 'A forest management planning model integrating silvicultural and transportation activities', *Management Science* **22**(12), 1299–1309.

Wolfe, P. (1963), 'A technique for resolving degeneracy in linear programming', *Journal of the Society for Industrial and Applied Mathematics* **11**(2), 205–211.

Yoshimoto, A., Brodie, J. D. & Sessions, J. (1994), 'A new heuristic to solve spatically constrained long-term harvest scheduling problems', *Forest Science* **40**(3), 365–396.

Glossary

.MPS A common file format for expressing a linear programming problem.

AC *Australian Company* One of the two companies that provided case studies.

Age-classes In Estate modelling a Crop-type is divided into age classes based on the year it was planted.

AGEN The subroutine in ZIP that is responsible for the initial generation of the constraint matrix.

ALLOC The subroutine in ZIP that is responsible for implementing the Integer allocation within the B&B.

Ambiguous location solutions Solutions to an OHS problem where crews' locations are not restricted in anyway and a single crew can harvest in many harvest units simultaneously.

Artificial variables Variables added to a LP problem that allow the initial non-singular Basis matrix to be formed as an identity matrix.

BANDB The subroutine in ZIP that is responsible for directing the B&B algorithm.

basic feasible solution A point in the Feasible region that corresponds to the solution of a Basis matrix.

Basis matrix A square non-singular matrix composed of variables in the LP problem.

B&B *Branch and Bound* A technique for finding integer solutions from a RLP.

Binary integer variable A variable in an IP or MILP, that may only be zero or one in a feasible solution.

Bound gap In B&B the difference between the best integer solution and the Objective value of the RLP solution.

Bound tolerance In B&B nodes are bounded when they are within the tolerance of the best integer value.

BRANCH The subroutine in ZIP that is responsible for deciding the appropriate branch and branch direction in the B&B.

Breadth-first branching Traversing a B&B tree by solving the best value nodes first.

Bucking Cutting a Stem into logs.

Bucking algorithm An algorithm that converts Standing inventory information into a Yield prediction using a particular Cutting strategy.

Bucking optimisation A similar problem to OHS but only considering allocation of production.

Bucking pattern A specific instruction on how to buck a particular Stem.

Bulldozer A heavy machine used to extract Stems.

Buy back When the forest company agrees to purchase back logs not needed by a customer.

CA *Crew Allocation* The decision to place a particular crew in a harvest unit in a period, also the constraints that model this decision in the OHS formulation.

Cable crews A Harvesting crew that extracts Stems by using Cable-logging, suitable for steeper country.

Cable-logging The extraction of logs using aerial cables.

Constraint branching A method used to find integer solutions to SPP problems.

Central processing yard A very large area where whole Stems from the forest are transported and bucked.

CGA *Column Generation Algorithm* The sub-problem in Column generation.

Column generation A technique used to find the solution to large LP problems by generating variables ‘on-the-fly’.

Clearfell A harvesting Operation where all the trees in a Harvest unit are felled

Clearwood Wood without defects, usually the result of a pruning regime.

Compartment A division of a Stand.

Complementary columns A set of Crew schedules which combine to give an unambiguous placement of a crew.

Complementary strategies A set of Cutting strategies that share the same log products.

Crew allocation The decision to place a particular crew in a Harvest unit in a period.

Crew movement sequence The sequence of movements of a crew throughout the time horizon in an OHS solution.

Crew schedule A sequence of crew allocations that shows a crew's movements from the beginning to the end of the planning horizon.

Crew-harvest unit branch A generalised constraint branch that allocates a harvest unit to a crew.

Crew-strategy branch A generalised constraint branch that allocates a cutting strategy to a crew and harvest unit.

Crop-type The aggregated land unit used in Estate modelling.

Cruising The collection of physical data from the trees within an area, typically by skilled forestry workers who walk through the forest and measure trees.

Cutting strategy Rules on how to determine a Bucking pattern given a particular Stem's features. Is used to determine Yield predictions.

Dantzig-Wolfe decomposition A decomposition algorithm for LP problems very similar to Column generation.

Degeneracy When a Basis matrix contains many variables at zero, is associated with cycling in the simplex iterations.

Delivered sales Log sales where, the customer receives the logs and pays a single price while the supplier pays all of the harvesting and transportation costs.

Depth-first branching Traversing a B&B tree by solving the deepest nodes first.

DP *Dynamic Program* A technique used to solve some types of OR problems.

Estate modelling The long-term forest planning problem, also commonly called Strategic planning.

End-effects Abnormalities in the solution of a model due to the truncation of the number of periods modelled.

Entering variable The variable that enters the Basis matrix on a pivot in the Simplex algorithm.

Extreme point A vertex of the Feasible region, corresponds to a basic feasible solution.

Faller A forestry worker that cuts down trees.

Feasible region In a LP problem, the solution space where all constraints are met.

Final stocking The number of trees per hectare left at harvest.

FOLPI *Forestry Orientated Linear Programming Interpreter* Estate modelling tool created at *forest research* and commonly used in NZ.

Forest gate sales Log sales where, transport costs are split between the supplier and customer. The supplier pays the portion to the edge of the forest.

FORTRAN A programming language.

Forwarder A heavy machine used to extract Stems.

forest research *New Zealand Forest Research Limited* The New Zealand Crown Research Institute responsible for research in forestry.

FSC *Forest Stewardship Council* An organisation that encourages sustainable forestry.

GIS *Geographical Information System*

Green-up constraints Common in Tactical planning these limit the harvest of neighbouring Harvest units until a clearfelled Harvest unit has begun to regrow.

Ground-based crews A Harvesting crew that extracts Stems by using heavy machinery to drag them along the ground, suitable for flat country.

GSPP *Generalised Set Partitioning Problem* An extension of the SPP.

GUB *Generalised Upper Bound* A term used to refer to a type of constraint common in scheduling problems.

Harvest unit integer feasible A property of a period that complies with a relaxed integer solution.

Harvesting crews A working group (in New Zealand (NZ) usually an independent contractor) that fells trees, extracts Stems, and makes logs.

Hierarchical planning The practice of dividing planning operations in separate elements that cover different decisions and time scales, see Strategic planning, Tactical planning and Operational Planning.

Harvest unit An area within a forest that is to be harvested in a single operation.

IFRLogger™ A instrument used to aid the Log making process.

Initial stocking The number of trees per hectare planted.

Integer allocation When a simple heuristic finds a good quality integer solution from a nodal solution, within the B&B.

Integer period A period that is required to be harvest unit integer feasible and strategy integer feasible.

Inventory plot A sample of a Stand used to determine Standing inventory measurements.

IP *Integer Program* An extension of an LP which only contains integer variables.

Landing An area of flat cleared land where log making and bucking occur, also known as a skid site.

Leaving variable The variable that enters the Basis matrix on a pivot in the Simplex algorithm.

LED *Large End Diameter* The diameter of the largest end of a log.

LED *Small End Diameter* The diameter of the largest end of a log.

Log allocation The process of allocating production within a forest to the market.

Log making The process that determines how a Stem will be Bucked into logs. Occurs before Bucking.

Log-scaling The techniques used to measure log volume.

Log-type A specific type of log with quality and dimensional specifications, for a particular customer and or use.

Logs Sections of Stem. Usually cut to sell to customers.

LP *Linear Program* A technique used to solve some types of OR problems.

Log-stocks Logs that have been harvested but not yet sold.

Market price The price a customer is willing to pay for a log.

MARVL *Method of Assessment of Recoverable Volume by Log-type* Standing inventory tool created at *forest research* and commonly used in NZ.

Mechanical feller A machine that fells trees.

MILP *Mixed Integer Linear Program* An extension of a LP to include integer variables.

Model I A possible formulation of the OHS problem. Commonly used in literature.

Model II A possible formulation of the OHS problem. Used in this thesis.

Model III A possible formulation of the OHS problem.

Model IV A possible formulation of the OHS problem.

Model V A possible formulation of the OHS problem.

Monte-Carlo simulation A type of optimisation meta-heuristic used for solving difficult combinatorial problems, uses random generation of initial conditions.

NFP *Network Flow Problem* A special subset of LP problems that will give naturally integer solutions.

NODE The subroutine in ZIP that is responsible for deciding the node to branch on in the B&B.

NZ *New Zealand*

NZC *New Zealand Company* One of the two companies that provided case studies.

Objective function In a LP problem, the function that determines values the points in the Feasible region.

Objective value The values of the Objective function at a particular point.

OHS *Operational Harvest Scheduling* Short term scheduling of forestry crews within a forest with production allocation.

OHSA *Operational Harvest Scheduling Algorithm* The algorithm that solves the OHS problem.

Old-crop Term used to refer to *Pinus radiata* that was planted circa 1920-1935.

Operational Planning Short-term detailed planning of actives to make the operation function.

Optimal solution In a LP problem, the point in the Feasible region where the Objective function is maximised.

OR *Operations Research* The science of better.

PEVAR The subroutine in ZIP that is responsible for finding the entering variable within the Simplex algorithm.

PLE *Probable Limits of Error* The confidence interval of an estimate expressed as a percentage of the mean.

PRIMAL The subroutine in ZIP that is responsible for using the primal Simplex algorithm.

Production thinning Thinning where the removed Stems are sold as various products.

Pruned Butt The first valuable log produced from the base of a Stem.

Prunning The removal of branches from the Stem of a tree to improve the wood quality.

PT *Production/ Transportation* The constraints that model the linear transportation, and production decisions in the OHS formulation.

RC *Reduced Cost* A measure of the improvement caused by the inclusion of a variable in the basis.

Regime A Program of silvicultural practices applied over the lifetime of a tree.

Relative prices The prices given to a DP bucker to generate a Yield prediction.

Relaxed integer solution Unrestricted movement solutions achieved by altering the requirements of an integer solution to a MILP.

Restricted movement solutions Solutions to an OHS problem where crews are unambiguously located in a single harvest unit in each period.

RLP *Relaxed Linear Program* The problem resulting when the integer restrictions on the MILP are removed, and the MILP solved as a standard LP problem.

RMP *Restricted Master Problem* The master problem in Column generation.

Roadside sales Log sales where, harvesting costs are paid by the supplier and the customer pays the transportation costs.

Rolling horizon models A model of a continual process, where only the first period is implemented. In the next period the model is resolved with updated data.

Rostering The assignment of subsets of jobs to entities that can complete them. For example, airline crew rostering.

Schedule A crew schedule without the crew, a sequence of Harvest units and periods.

SED *Small End Diameter* The diameter of the smallest end of a log.

Set-covering problem Similar to the SPP but with $\geq$ constraints.

Set-packing problem Similar to the SPP but with $\leq$ constraints.

Simplex algorithm A method used to solve LP problems.

Standing inventory Also simply Inventory. The trees that are in the forest before they are felled.

Skidder A heavy machine used to extract Stems.

Sparsity A property of a matrix where most of its elements are zero.

Strategic planning Planning that defines the role and nature of the enterprise. In forestry this term is commonly used to refer to extremely long term Tactical planning, see also Estate modelling.

SPP *Set Partitioning Problem* A special type of IP.

Stand Separate physical locations within the forest that are internally homogeneous with respect to future management.

Stem A fallen tree without branches.

Stem-class A set of Stems with identical properties.

Stem-description A physical description of the dimensions and quality of a Stem.

Strategy integer feasible A property of a period that satisfies the integer requirements for cutting strategies.

Stumpage sales Log sales where, logs are sold within the standing tree, harvesting and transport costs are paid for separately by the customer.

Super-skid A large landing that services a number of harvesting operations commonly used in Cable-logging operations.

Tabu search A type of optimisation meta-heuristic used for solving difficult combinatorial problems, uses a list of banned (Tabu) movements to move between local optima.

Thinning The removal of trees before the final harvest.

Timber Boards, fittings etc., that are the products of sawmills.

Tactical planning Planning to make the most effective use of the enterprise's resources in the medium-term.

Tree A tree within a forest before it is felled.

UNPACK The subroutine in ZIP that constructs the matrix column that represents the entering variable.

Unrestricted movement solutions Solutions to an OHS problem where crews are unambiguously located in a single harvest unit at any time within the time horizon. However, the crews may also move between harvest units within a period.

Variable branching A commonly used method to find integer solutions to IP, or MILP problems.

Wofle's method A technique for overcoming Degeneracy problems in the Simplex algorithm

Yield prediction subproblem The subproblem which generates new yields based on the requirements of the master problem.

Yield prediction The predicted log volumes from a Harvest unit.

ZIP *Zero-one Integer Programming* A programming framework developed at the University of Auckland (Ryan 1980), for the solution of large scheduling problems.