Optimization with a Highly-Constrained Product Mix

Howard A. Leach, HALCO Software Systems Ltd.

Introduction

The problem discussed here is how to determine which sawing patterns should be allowed for each log, and what product prices and penalty costs for machine use should be used in sawmill optimizers, when the volume of each product that should be manufactured is determined largely by market demand rather than price.

Under these circumstances it is the demands for products, not the selling prices, that are the primary drivers of which sawing patterns should be used for which log. However, most sawmill optimizers use product prices as the primary driver when choosing between sawing patterns.

There is usually some flexibility in how much of each product can be sold, so that if a little more of products that are in demand and can command a higher price can be made, and a little less of products that are in less demand and command a lower price can be made, then the mill may be more profitable. Regardless of this, some sawing patterns fit the products into the log better, yielding a higher product recovery and a lower log cost per unit of product volume, and some make more efficient use of the equipment, resulting in a higher production rate and either a lower production cost per unit of product volume or increased total volume of production and sales. So the highest valued products do not necessarily yield the highest profits. There is still plenty of scope for optimization!

The purpose of this document is to describe an optimization procedure that is a combination of a rule-based decision-making process, combined with manipulation of product prices and penalty costs for use of machine time. An attempt will be made to explain why this procedure is likely to produce much better results than one which is based mainly on manipulating product prices.

Conventional Optimization Procedure

The conventional on-line optimization procedure is to determine in advance a set of product prices to be used in the optimizers (typically determined from finished-product prices), to use arbitrary penalty costs for use of machines that may limit production, and to set up a series of arbitrary rules to determine which center-cant and side-board thicknesses will be considered for logs of various diameter and length classes. Normally it is necessary to make additional arbitrary adjustments to the green-end product prices in order to control the product mix. One way to do this is to tally the accumulated productions of each product size, and then to adjust the optimizer values in order to steer the accumulated productions to match the desired levels. In effect a negative feed-back loop is created, with the attendant risk of instability in production rates for individual products.

Constrained Optimization Procedure

The constrained optimization procedure described here first uses a linear programming model to prepare a production plan for a production cycle (perhaps 10 days of production), during which the full range of log lengths will be processed. For planning purposes the logs expected to be processed during a production cycle are divided into diameter, length and possibly quality classes, and the expected product yields and machine-time requirements for each log class processed with all possible sawing patterns are calculated in advance. The linear program is then be used to determine:

- which pattern or combination of patterns should be used for each log class (length, diameter and quality class),
- the product mix expected to be produced from each log class,
- the marginal breakeven values of each rough green product,
- the marginal costs of time on each green- and dry-end machine,
- the expected total production of each product.

The production is constrained in the model to match the product requirements and the capacities of the various green- and dry-end machines.

An automated iterative procedure is used to fine-tune the marginal breakeven values to be used in the optimizers to obtain an improved solution.

The problem with this, by itself, is that not enough is known in advance, before each log is scanned, to determine exactly what products and machine time requirements will result when a log is sawn by a particular pattern. So there is plenty of scope for an on-line optimizer to improve on the pre-calculated sawing decisions on a log-by-log basis.

However, in a conventional implementation, the on-line optimizer does not know what logs are expected to be "coming down the pipe". Some log classes required to meet certain "hard-to-make" products may be in short supply. So the only way to prevent these log classes from being used to make "easy-to-make" products is to implement some rule-based procedure. This is done by specifying to the on-line optimizer, in advance, the product mix expected from each log class, the sawing patterns that are candidates for that log class, and the marginal breakeven values for products and machine time, as calculated by the linear program.

Then, in order to match the product mix, the on-line optimizer needs to use the values and machine costs in its optimization (of rotation, log alignment, pattern offsets, sideboard "fill", and gang offset) to calculate the product yields, machine-time requirements and total value for each candidate sawing pattern. The optimizer would then exclude solutions whose value differs from the maximum-value solution by more than a pre-determined amount. Then it must select from the remaining patterns the one that will make the maximum contribution to closing the gap between the accumulated production for that log class and the desired production assigned by the linear programming model for that class.

Why is the constrained optimization procedure superior to the conventional method?

1. The product-mix matching procedure proposed for the on-line optimizer in the constrained optimization procedure has been used for bucking optimizers, and at least one primary breakdown installation, and is known to work very well indeed.

2. There are theoretical reasons why the constrained procedure's rule-based method, combined with values for products and costs for machine time, should work better than a value-based feed-back loop.

Theoretical reasons for the constrained procedure's rule-based method

The easiest way to explain these is through a discussion of how the simplex method of linear programming works.

In a linear-programming model of a sawmill, variables are defined that represent the number (or volume) of logs in each class that will be sawn by each sawing method (a sawing method being a group of similar sawing patterns, perhaps with some sub-optimization). There may be additional variables to represent log-yard and finishing operations, and the total volumes of logs and time used and products made.

An objective function is set up to represent the net income from the operations, and there must be at least one constraint to prevent an unlimited number of logs from being processed. The components of the objective function are the values of the products and the costs of logs and mill time.

If there were no constraints on product volumes and machine times, a manual solution would be simple. Just select the sawing methods which yield the maximum net income and saw all the logs that are available. In other words, maximise the return-to-log.

In practice, there must be additional equations (actually inequalities) to represent the constraints on the amounts of logs and machine time available, and particularly in situations where product volumes must be determined largely by market demand rather than price, it is necessary to have many additional constraints to represent limitations on the product volumes that can be sold.

When there are more than a very small number of additional constraints, a manual solution rapidly becomes impossible. It is certainly not possible to successfully implement either a manual or computerized procedure where the product prices are artificially manipulated so that maximizing the revised objective function will result in a choice of sawing solutions that will satisfy the constraints. Changing the prices so that one constraint is satisfied invariably means that another constraint is not satisfied, and fails when there are more than a few constraints.

The basic problem is that one price for a product is applicable until a sales constraint is reached, and beyond that a particular product may have no value at all. Adjusting the price by trial and error in the hope that just the right amount of a product will be produced does not work well, particularly with there are more than a few such constraints.

A linear program works by finding a solution that maximises value until a single constraint is met, and then sliding along another constraint until two constraints are met, and so on for additional constraints. It is a programmed procedure rather than a trial-and-error approach.

The method suggested here is somewhat similar: find a number of solutions that are near-optimal, and then choose the one that takes the solution in the required direction. It is more of a programmed procedure than a trial and error approach.