An Abstract Of The Thesis Of

Peter D. Eldred for the degree of Master Of Science in Forest Economics presented on November 25, 1987.

Title: Evaluation Of A Shadow Price Search Heuristic As An Alternative To Linear Programming Or Binary Search For Timber Harvest Scheduling

Abstract approved: ______________________________
Norman K. Johnson

The planning of harvests and management activities for forested lands has traditionally been done with either binary search or linear programming. Since both these techniques have some advantages over the other, they have remained in wide use. Hoganson and Rose (1984) have suggested a technique that theoretically could overcome some of the problems with binary search and linear programming while retaining many of their desirable characteristics. The technique uses shadow prices to guide the search for harvest levels in each period to meet certain goals. This study develops a new set of shadow price search procedures and makes a comparison between these three different harvest scheduling techniques.

The comparisons were made by constructing a PASCAL computer model that gives the user a choice of solving

harvest scheduling problems with either binary search or this new method, that will be referred to as "shadow price search". The computer program MUSYC was used for the linear program formulations. Three example were solved by each of these methods.

The results showed that linear programming and shadow price search produced solutions with similar harvest patterns among stands and present net worths that were close in value. Binary search found very different harvest patterns and consistently lower present net worths than either of the other two methods. Solution times for shadow price search were greater than binary search, but still less than linear programming solution times.

Shadow price search seems a promising alternative to the traditional approaches for harvest scheduling problems with many timber stands and few constraints. For these problems, it's major drawback is occasional difficulty in converging on the optimal solution. Future research may solve this problem.

EVALUATION OF A SHADOW PRICE SEARCH HEURISTIC AS AN ALTERNATIVE TO LINEAR PROGRAMMING OR BINARY SEARCH FOR TIMBER HARVEST SCHEDULING

By

Peter Eldred

A Thesis

Submitted to

Oregon State University

In partial fulfillment of

the requirements for the

degree of

Master of Science

Completed november 25, 1987

Commencement June 1988

Table Of Contents

List Of Tables

Evaluation Of A Shadow Prices Search Heuristic As An Alternative To Linear Programming Or Binary Search For Timber Harvest Scheduling

Introduction

Public and private firms have become increasingly dependent on mathematical modeling for forest management decisions. These models help decide the land to allocations for timber production, the harvest schedule, and the timing of silvicultural activities. The models have generally been of two types: linear programming (TIMBER RAM, Navon, 1971) (MAXMILLLION, Clutter et al., 1968) (MUSYC, Johnson and Jones, 1979) (FORPLAN, Johnson et al., 1986), and binary search (SIMAC, Sassaman et al., 1972) (ECHO, Walker, 1976) (TREES, Johnson et al. 1975) (Tedder et all 1980).

Linear programing is able to handle multiple constraints and determine the optimum management intensity as part of the solution. Flexibility in the definition of the objective function, along with the capacity to specify many constraints, gives linear programming the capacity to solve complex forest planning problems. If there is a feasible solution given the constraints, linear programming will converge on the optimal solution. Unfortunately, harvest scheduling problems with many stands and constraints become extremely large and complex to formulate and solve with linear programming. Also, the results can be difficult to interpret.

Binary search is a simple algorithm that can be solved relatively quickly even for forests with many inventory catagories. It is a heuristic approach as it just follows a set of logical steps until it converges on a reasonable solution. It is not a true mathematical optimizer. Binary search relies on a preset priority such as "oldest timber first" or "slowest value growth first" to decide which stands to harvest. The speed and ease of binary search has made it a popular alternative to linear programming for harvest scheduling problems, but it has limitations. The number of constraints that can be applied is very limited. Also, predetermining the priority for harvest and designating the management intensity for each stand reduces the likelihood of arriving at an optimal or near optimal solution (see Johnson and Tedder 1983 for more discussion of linear programming Vs. binary search).

Hoganson and Rose (1984) have presented a new heuristic for harvest scheduling problems that uses the principle of linear programming decomposition to find integer solutions. The hope is that this new technique, that will be referred to as a "shadow price search", will retain much of the solution speed of binary search while producing a result with a harvest pattern and a present net worth similar to that of linear programming.

This study is designed to test the viability of the shadow price search algorithm as an alternative to the traditional means of harvest scheduling. The basic mathematical technique is presented along with the variations that were used to implement it. Three harvest scheduling examples are solved with all of these techniques. The comparison of the solution times and the resulting harvest schedules reveal some of the relative strengths and weaknesses of shadow price search relative to linear programming and binary search.

Objectives

This study provides information to forest managers for choosing the best management tool to use for harvest scheduling on their forest lands.

The harvest scheduling methods of binary search, shadow price search and linear programming are compared. The evaluation criteria used will be: 1) ease of use, 2) present net worth of the solution, 3) degree of convergence on harvest goals, and 4) the limitations the techniques have with respect to solving various types of harvest scheduling problems. These results, along with a comparison of the age classes harvest per period (harvest patterns), will be used to evaluate the shadow price search heuristic as an alternative to linear programming or binary search.

Harvest Scheduling Using Linear Programming Decomposition

With linear programming, volume or value can be maximized subject to many constraints, including available acres, harvest volume, acres harvested per period, age class distribution per period, and the amount of inventory to be left at the end of the planning horizon. Also, constraints can be applied to cost and revenue levels. In this study, only available acres and harvest flow constraints were applied to the objective of maximizing present net worth.

When a harvest scheduling problem is formulated as a linear program, two basic mathematical structures can be used, model I or model II (Jonson and Scheurman 1977). With a model I structure, stands maintain their integrity throughout the planning horizon. A model II structure lumps stands of the same type together after harvest. Formulating a problem with a model II structure reduces the number of decision variables that need to be specified, as stands can be combined after harvest, but increases the number of constraints. The optimal solution remains the same whether model I or Model II is used. Both formulations were used in this study for reasons discussed in the procedures chapter.

The shadow price search heuristic takes advantage of a technique called decomposition (Dantzig and Wolf 1961) to solve harvest scheduling problems by breaking them into smaller subproblems. Dantzig-Wolf decomposition exploits the structure of block angular linear programs. Block angular problems consist of a group of independent subproblems that have a set of coupling constraints linking them together. Any point in the feasible region of any subproblem can be defined in terms of a convex combination of the finite extreme points of the subproblem. A new problem is formulated in terms of the extreme points of the subproblems, with the decision variables being the weights, or shadow prices, placed on each of the extreme points. This reformulated problem is called the master problem. The smaller subproblems are identified and optimized independently for a given set of shadow prices. The optimal solutions from the subproblems determine the next set of variables, or extreme points, to enter the basis in the master problem. When the master problem is solved, a new set of shadow prices is found and passed back to the subproblem. The subproblems are then optimized with the new shadow prices. This process of alternately solving the master problem and the subproblems is continued until no improvement in the objective function can be made.

The advantage of decomposition is that large, hard to solve, problems are broken down into smaller units that can be solved relatively easily. Also, only the coefficients for the decision variables chosen to enter the basis by the subproblems need to be calculated.

Model I harvest scheduling problems exhibit this special block angular structure. In the examples used in this study, each unique stand is a subproblem and the linking constraints are the harvest level goals. The decision variables are the management options for each stand. Littschwager and Tcheng (1967) used decomposition to maximize wood production from 1166 cutting units. There were no harvest constraints, instead the linking constraints were an upper and lower bound on the number of acres harvested each year.

The general formulation for Dantzig-Wolf decomposition applied to harvest scheduling problems was presented by Nazareth (1980). The suggested solution technique was the revised simplex method for the master problem and dynamic programing for the subproblems. Rather than solving a specific example, just a general formulation was presented for land management models.

Berck and Bible (1984) used a dual approach with Dantzig-Wolf decomposition to formulate harvest scheduling problems with only area and harvest flow constraints. The revised simplex method was used to solve the master problem. The master problem was a model I formulation and the subproblems model II formulations.

All of the previously mentioned studies used the simplex method for the solution of the master problem. Hoganson and Rose (1984) used a heuristic dual decomposition approach. The general framework for this approach was presented by Everett (1963). Everett referred to block angular linear programming problems as "cell" problems, meaning they could be broken into independent subproblems. He used the Lagrange multiplier technique to convert constrained maximization problems to unconstrained maximization problems. Lagrange multipliers are also referred to as dual variables or shadow prices. The standard optimization method for Lagrange multipliers is to apply the first order conditions for maximization, finding the point where the slope of the function to be optimized equals zero. This is done by setting all the derivatives of this function equal to zero and solving. Everett suggested that the Lagrange formulation could also be solved by trying different values for the Lagrange multipliers until the constraints were met. This guessing solution technique is more versatile, as it is not confined to functions that can be differentiated. Everett gave little insight as to how this guessing process would efficiently proceed.

Brooks and Geoffrion (1966) suggested using linear programming to find Everett's Lagrange multipliers. With this technique, simplex code is used to solve the primal problem in a process similar to standard Dantzig-Wolf decomposition. Geoffrion (1974) referred to this technique as "Lagrange relaxation" and suggested other techniques for adjusting the Lagrange multipliers, such as tangential approximation. This method, along with subgradient algorithms (Bazarra and Sherali 1981) (Held, Wolfe and Crowder 1974) and gradient algorithms, (Lasdon 1970,1972) have a mathematical basis for convergence on the optimum values. They also have a significant computational burden. The additional computation time may partially or totally offset the advantage of using Lagrange multipliers over traditional linear programming. The primary use for these algorithms has been to find integer solutions, which is a time consuming process if done by the traditional integer programming method of branch and bound.

Hoganson and Rose (1984) developed procedures to search for shadow prices without using simplex code or a mathematical basis for convergence. They applied this method to several harvest scheduling problems. This totally heuristic approach for arriving at the shadow prices appears to be unique among forestry applications. This study develops new procedures for adjusting the shadow price estimates, and makes a comparison with other traditional solution methods.

Problem Formulation

When using the shadow price search approach, one of two following assumptions must be made: 1) the constraints are somewhat flexible, so they do not need to be met exactly, or, 2) finding an integer solution, where an entire inventory category is assigned to a management strategy, is more important than exactly meeting the constraints. This technique breaks the forest wide harvest scheduling problem into individual stand optimization subproblems. A model I formulation is used, with stands retaining their integrity throughout the planning horizon. The primal harvest scheduling problem can be presented as:

$$\text{Max:} \quad \sum_{i=1}^{I} \sum_{j=1}^{J_i} c_{ij} x_{ij} \qquad (1)$$

$$\text{ST:} \quad \sum_{i=1}^{I} \sum_{j=1}^{J_i} v_{ijt} x_{ij} = M_t \quad \text{for all } t$$

$$\sum_{j=1}^{J_i} x_{ij} \le A_i \quad \text{for all } i$$

$$x_{ij} \ge 0 \quad \text{for all } i,j$$

Where:

A_i=Acres of stand of type i

c_{ij}=The present net worth of assigning an acre of stand type i to the management sequence j

M_t=The desired harvest level in period t

I=The number of stand types

J_i=The number of management sequences for stand type i

v_{ijt}=The harvest volume per acre in period t for stand type i with sequence j

x_{ij}=Acres of stand type i assigned to sequence j

This formulation maximizes present net worth of timber harvest subject to harvest flow and acreage constraints. The harvest flow constraints are the harvest goals (M_t) for each period. Acreage constraints (A_i) simply state that the acreage harvested can not be more than what is available. Converting this problem to the dual formulation results in the following formulation:

$$\text{min: } \sum_{t=1}^{T} M_t m_t + \sum_{i=1}^{I} A_i a_i \qquad (2)$$

$$\text{ST: } \sum_{t=1}^{T} v_{ijt} m_t + a_i \geq c_{ij} \quad \text{for all } i, j$$

$$a_i \geq 0 \quad \text{for all } i, \quad m_t \text{ unrestricted}$$

where:

m_t=Dual variable for M_t

a_i=Dual variable for A_i

If it is assumed that m_t is known, then the first expression in (2) becomes a constant, and the dual problem can be written as:

$$\text{min: } \sum_{i=1}^{I} A_i a_i \qquad (3)$$

ST: $\sum_{t=1}^{T} v_{ijt}m_t - c_{ij} \geq -a_i$ for all i, j

This constraint can be written as:

$$c_{ij} - \sum_{t=1}^{T} v_{ijt}m_t \leq a_i \quad \text{for all } i, j \qquad (4)$$

This last equation (4) is the decomposed subproblem that will be solved for each stand. Note that there are "j" constraints for each stand type, each with the same a_i right hand side. The only way to satisfy all "j" constraints for a stand type is for a_i to be greater than the largest of the left hand sides. Referring to equation (3), it can be seen that the optimal solution will contain the smallest possible a_i's that satisfy equation (4), so a_i will equal the largest left hand side. This means that the optimum a_i values can be found by solving for the maximum left hand side. Complementary slackness confirms this result. By complementary slackness, if optimality exists and a x_{ij} primal variable is positive, then the corresponding dual constraint is met by an equality. In this case, a_i equals the left hand side.

For each iteration of shadow price search the dual master problem is solved for a given set of shadow prices. Since we are starting with the dual problem, all the solutions found will be dual feasible. By adjusting the shadow prices (m_t) for the per period harvests we can work towards, but probably not attain, primal feasibility.

It is important to note that the problem being solved by shadow price search is not precisely the same as the primal formulation. Because of the way the shadow price search algorithm proceeds, only integer dual solutions are found, while the primal problem is continuous. An integer formulation of the original primal problem would be as follows :

Max: $$\sum_{i=1}^{I} \sum_{j=1}^{J_i} c_{ij} [\underline{x}_{ij} A_i]$$

ST: $$\sum_{i=1}^{I} \sum_{j=1}^{J_i} v_{ij} [\underline{x}_{ij} A_i] = M_t \qquad \text{for all } i$$

$$\sum_{j=1}^{J_i} [\underline{x}_{ij} A_i] \leq A_i \qquad \text{for all } i$$

$$\underline{x}_{ij} = [0,1]$$

where:

$\underline{x}_{ij}$=The proportion of acres of stand type i assigned to sequence j

All other variables are the same as defined previously.

The dual formulation of this primal problem has exactly the same objective function as the previous dual formulation. The constraint is as follows:

$$\sum_{t=1}^{T} v_{ijt} A_i m_t + a_i A_i \geq c_{ij} A_i \quad \text{for all } i,j$$

Dividing both sides of this constraint by "A_i" results in the following inequality:

$$\sum_{t=1}^{T} v_{ijt} m_t + a_i \geq c_{ij} \quad \text{for all } i,j$$

Which is precisely the same as the previous dual formulation constraint, so the dual formulations of the continuous and integer formulations are equal. This supports using equation (4) to solve the integer problem.

The algorithm steps are :

1) Make an initial estimate at the shadow prices for harvest in each period (zero if you have no prior knowledge).
2) Solve the subproblems assuming the shadow prices (dual variables) are correct.
3) Check to see how close to primal feasible the solution is (i.e. how close to the desired harvest levels the per period harvest is).
4) If within the desired limits of primal feasibility, then stop, else re-adjust the shadow prices to move the solution closer to primal feasibility, then resolve the problem.
5) Repeat steps 2 through 4 until a "near feasible"

solution is found that is within the acceptable limits of deviation from the desired harvest levels.

Since the harvest regime for every stand is optimal, the forest wide solution is always optimal for whatever per period harvest levels result. If the initial shadow price estimates are all zero for the first iteration, the result is the unconstrained maximum present net worth. This can be seen by looking at the left-hand side of equation 4. If the shadow price is zero, the left-hand side becomes simply c_{ij}, the present net worth of assigning an acre of stand "i" to management sequence "j". Maximizing this present net worth for every stand will result in the unconstrained maximum present net worth for the entire forest.

Each iteration, a search is made through the possible harvest sequences for each stand to find the one that maximizes the value of the decomposed subproblem for the given the shadow prices and discount rate. The first component of equation (4), c_{ij}, can be thought of as the unconstrained value of the harvest sequence. the second component, the rest of the left hand side, can be thought of as the cost of the constraint. If the shadow price is positive, the second component becomes positive, representing a positive constraint cost. As would be expected, this positive cost tends to force harvest out of that period. If the shadow price is negative, the second component becomes negative, representing a negative cost, or a benefit. This has the effect of drawing harvest into the period. The trick is to effectively adjust the shadow prices, so when the subproblems are solved for optimality, the harvest levels will come close to the desired levels.

Shadow price adjustment Procedures

Hoganson and Rose (1984) used three techniques to adjust the shadow prices, 1) "float", 2) "shape", and 3) "smooth". These adjustment procedures are based on the principle that the shadow prices over time are a continuous smooth curve. "Float" moves the entire shadow price curve up or down by a given amount depending whether the total harvest over all the planning periods is higher or lower than the desired level. "Shape" moves each period's shadow price either up or down depending on the deviation from the harvest goal in that period and the surrounding periods. Doing this alters the shape of the shadow price curve over time. "Smooth" makes fine adjustments in the slope of the shadow price curve, looking at two periods at a time. By varying the slope between planning periods "smooth" attempts to shift volume from periods with too much volume to periods with too little volume. These three procedures are used with different sequences and parameters depending on the problem, until a solution is found. It is up to the user to find the best sequence and parameters.

These procedures use harvest and shadow prices from adjacent periods to determine the new shadow price for each period. This assumes a relationship between periods which may not exist. In an example with large jumps in shadow price between periods, these techniques can have difficulty converging on the proper harvest levels. A jump in shadow prices between periods is not unusual and can be caused by a number of factors. A shift to a different species with a different value, a shift to harvesting different age groups, such as a shift from old growth to new growth, or a harvest goal much higher or lower than would naturally be available in that period can all cause abrupt changes in shadow prices.

Due to this problem, three new shadow price adjustment procedures were created, 1) "rough-price", 2) "new-price", and 3) "default-price". These procedures are shown below:

Rough-price

$m_{ti}=m_{ti-1} + b(adj_{ti})$

Each time the shadow price adjustment changes direction $adjt_i$ is reduced by 20 percent.

New-price

$$m_{ti}=m_{ti-1} + b\ (m_{ti-2} - m_{ti-1})(H_{ti-1} - M_t)/(H_{ti-2} - H_{ti-1})$$

Default-price

$$m_t=m_{ti-1} + c(H_{ti-1} - M_t)/M_t$$

where:

m_{ti}=Shadow price for harvest in period t in iteration i

M_t=Desired harvest for period t

H_{ti}=Actual harvest for period t in iteration i

adj_{ti}=amount of change for period t in iteration i

c=constant to adjust magnitude of change

b=1 if the previous harvest level was too high, or

-1 if the previous harvest level was too low

The initial adjustment value (adj_{ti}) is specified by the user. Each time the shadow price guess changes direction (increasing to decreasing or vice versa) the adjustment amount is reduced by one fifth, a fraction arrived at through trial and error. Each period has it's own adjustment amount, and the adjustments are varied independently of other periods. This way, a large adjustment can be done with a minimum number of iterations. "Rough-price" was used in this study until the maximum adjustment value of all periods was less than 0.05.

"New-price" is implemented after "rough-price" to make fine adjustments. It looks back over the last two iterations for the period being adjusted. A ratio is made of how much the shadow price was varied to how much the harvest changed. Then, assuming future harvest levels will react the same as past harvest levels to the amount of shadow price adjustment, a new shadow price is calculated to move harvest closer to the desired level. To compensate for the tendency to overcorrect for large deviations, the square root of the change is used when harvest deviations are high. As an example, say over the last two periods the shadow price was changed from -1 to -3, and the harvest went from 20,000 MBF to 30,000 MBF, and the harvest goal is 35,000 MBF. "New-price" would calculate that decreasing the shadow price by two resulted in a 10,000 MBF increase in harvest. Therefore, obtaining a 5,000 MBF increase in the harvest could be accomplished by subtracting one from the last shadow price resulting, in a new shadow price of -4. Chances are -4 would not produce a harvest of exactly 35,000 MBF, but it will hopefully move us closer to the answer. This procedure is used until the harvest is within a given deviation (specified by the user) from the desired levels.

"Default-price" is used when "new-price" cannot be used. This occurs when the harvest level or the shadow price has not changed in the last two iterations. In this case the computed difference in shadow prices or harvest levels over the previous two iterations is zero, and applying the "new-price" formula results in an adjustment of zero (shadow price difference zero) or an adjustment of infinity (harvest level difference zero). "Default-price" computes the percent deviation from the desired harvest and adds or subtracts a fraction of that amount to the old shadow price: 1/10 was the fraction used in this study.

Let us look at a simple three period example (table 1). The first shadow price guess is zero, and the harvest goal is 5,000 MBF. For this example, the initial adjustment value for "rough-price" will be ten, and the switch to using "new-price" procedure made when the absolute value of the maximum adjustment is less than 8.5, and termination will occur when the harvest levels are within ten percent of the harvest goals. In the second iteration (table 1) "rough-price" just adds ten to the shadow price if the harvest was too high in iteration one, or subtracts ten if the harvest in iteration one was too low. The result is period one harvest went from too much to too little, and vice versa in period two. This change of direction causes the adjustment amount to be reduced by 20 percent to eight for both periods one and two for iteration two. Note the adjustment amount for period three in iteration two stays at ten, since there was no change of direction. "Rough-price" is used until iteration five, when the maximum adjustment factor is eight, and the switch is made to the "new-price" procedure. Period one does not switch to "new-price" in the fifth iteration, as it's harvest level was the same in iterations four and five, so "default-price" is used instead.

Table 1 : Shadow Price Adjustment Example

Iteration		Period 1	2	3
1	harvest	10,000	4,000	1,000
	Shadow Price	0	0	0
	Procedure used	roughprice	roughprice	roughprice
	Adjustment	10	-10	-10
2	harvest	4,000	9,000	2,000
	Shadow Price	10	-10	-10
	Procedure used	roughprice	roughprice	roughprice
	adjustment	-8	8	-10
3	harvest	8,000	6,000	3,000
	Shadow Price	2	-2	-20
	Procedure used	roughprice	roughprice	roughprice
	adjustment	6.4	8	-10
4	harvest	4,000	3,000	7,000
	Shadow Price	8.4	6	-30
	Procedure used	roughprice	roughprice	roughprice
	adjustment	-5.1	-6.4	8
5	harvest	4,000	5,500	4,000
	Shadow price	3.3	-0.4	-22
	Procedure Used	defaultprice	newprice	newprice
	adjustment	-.02	-1.28	-2.67
6	harvest	5,350	4,600	5,000
	Shadow Price	3.28	0.88	-24.67

The calculations for the shadow price adjustments in iteration five are shown below.

Period 1 : <u>Default-price</u>

m_{16} = 3.3 + 0.1(4,000-5,000)/5,000

= 3.3 - 0.02 = 3.28

Period 2 : <u>New-price</u>

m_{26} = -0.4 + |(6+0.4)(5,500-5,000)/(3,000-5,500)|

= -0.4 + 1.28 = 0.88

Period 3 : <u>New-price</u>

m_{36} = -22 - |(-30+22)(4,000-5,00)/(7,000-4,000)|

= -22 - 2.67 = -24.67

After this iteration, the harvest is within the ten percent deviation, and the algorithm stops. If more iterations were needed, the "new-price" procedure would continue to be used.

The switches from using "rough-price", "new-price" and "default-price" are automatically implemented by the algorithm, so no additional user input is required once the process is started.

Procedure

The objective of all of the examples in this study was to maximize present net worth, subject to even flow of harvest volume and acreage constraints. For simplification, no minimum inventory conditions for the end of the planning horizon were specified and the only harvest option allowed was clear cut. All harvest were assumed to occur in the middle of the period, with a period length of ten years, and a seven period planning horizon. A four percent discount was used. The three procedures used to solve the problems were 1) binary search, 2) linear programming, and 3) shadow price search.

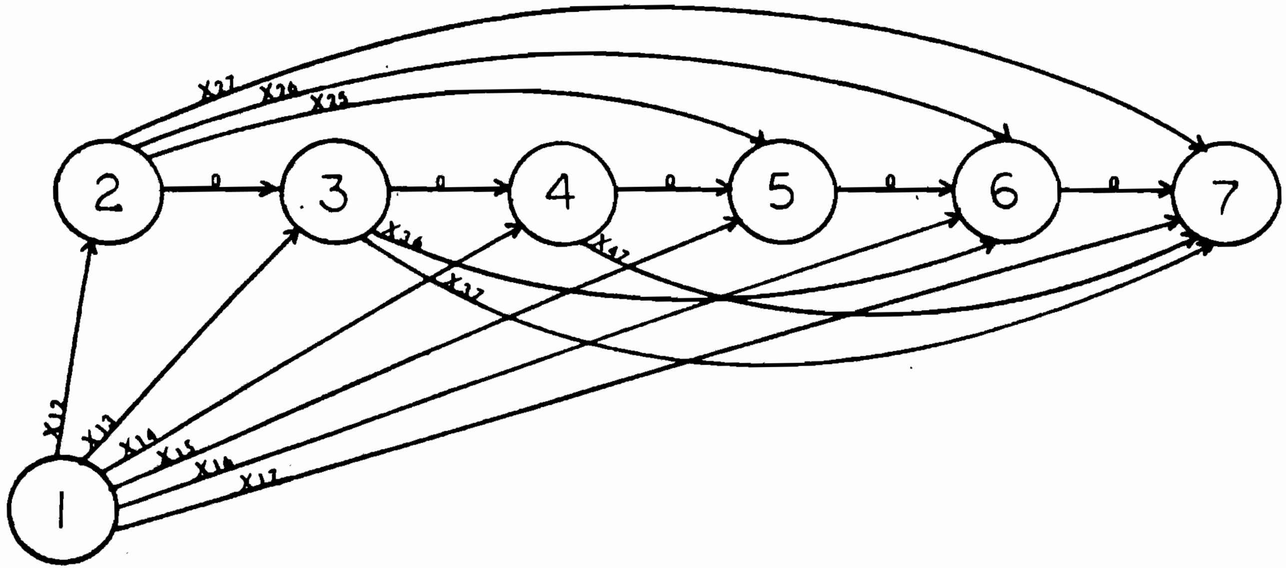
Binary search is a simple iterative procedure that attempts to find the maximum sustainable harvest level over all the planning periods. This search process can take several forms. One way is to start with an initial harvest level guess and a step size. The algorithm tries the initial harvest level. If it is too low, the harvest is increased by the initial step size, if the harvest level is too high, it is decreased by the initial step size. A predetermined rule for harvest priority, such as oldest growth first, or slowest growth first, determines the harvest pattern. A harvest level is too high if it can not be sustained through the final period and meet the ending conditions. A harvest level is too low if the harvest level can be sustained, but the ending conditions are exceeded by a specified amount of available harvest volume. Each time the harvest level guess changes from increasing to decreasing, or vice versa, the step size is halved. This process continues until the step size drops below some predetermined amount, or the desired ending conditions are met within specified limits.

A similar binary search algorithm starts with an upper and a lower bound on the harvest level. The midpoint between the bounds is used as the harvest level guess each iteration. If the harvest level guess is too high, it becomes the new upper bound, if it is too low, it becomes the new lower bound. In this way, the range of harvest levels being searched is halved every iteration, until the ending conditions are converged upon.

A PASCAL program was created to implement this second binary search algorithm (available at the department of forest management, Oregon State University, Corvallis Oregon). The initial upper bound was the maximum amount of timber that could be harvested in the first period, and the initial lower bound was zero. The program calculated the maximum first period harvest by adding up the current volume of all the management units. Using this second algorithm reduces the number of initial parameters to be specified by the user. A model II formulation was used with an "oldest first" harvest priority. An oldest first harvest priority is a common priority rule used for binary search models. The ending condition was not more than one half of one percent of the per period harvest level left after the final harvest. Binary search can not maximize present net worth, our stated objective, but by attempting to maximizing sustained volume using a "good" harvest priority rule, this objective is simulated to some degree.

A micro version of MUSYC (Johnson and Jones 1979) was used with LINDO (Linus Schrage, University of Chicago, 1981) to solve the linear programming formulation of the problem. MUSYC generates the mathematical matrix that LINDO solves using simplex code. A model II formulation was used to reduce the number of decision variables. As previously stated, using model I or II does not affect the optimal solution.

The shadow price search algorithm was also implemented with a PASCAL program (available at the department of forest management, Oregon State University, Corvallis, Oregon). A model I formulation was used. This is the natural way to apply shadow price search because of the way the problem is decomposed into individual subproblems for each stand. Dynamic programming was used to solve the subproblems. In the examples used in this study there were no intermediate harvests, only final harvests, so a simplified dynamic programming formulation could be used as shown (diagram 1).



<u>**Diagram #1**</u>

Dynamic programming formulation

7 periods with a minimum harvest age of 30 years, 10 years between periods.

X_{ij} = The net value of harvesting in period j when the current period is i.

Shadow price search requires an initial harvest goal, so binary search harvest levels were used as a starting point with Initial adjustment values of 10 or 12 for adj_{ti}. Shadow price was run for 50 iterations, then the average harvest per period was used as the new harvest goal, and shadow price search rerun. The solution times given for the shadow price search runs are the sum of the binary search and the two shadow price search runs. The termination condition in all the shadow price search runs was less than a ten percent deviation, plus or minus, from the harvest goal in each period.

All the programs were run at 16 MHz on a COMPAQ 386 with an 80387 math co-processor.

The general data requirements are:

1) Existing timber stands with information on acreage and initial average ages.
2) Existing stand harvest tables containing the volume, price, and cost of timber by age class.
3) Future stand harvest tables containing the volume, price, and cost of timber by age class. The initial and future harvest tables may be the same.

Two basic data sets were constructed. The first was a hypothetical set of 95 stands (table 7 in appendix). There were five different types of stands, 19 stands in each type. The initial age of the stands ranged from ten to 190. Each type had it's own initial table and regenerated table for volume and value at each age, so there were five initial and five regenerated harvest tables (tables 8-12 in appendix). The minimum harvest age was 30 years.

The second data set was actual inventory data from a 1976 Virginia forest inventory conducted by the U.S. Forest Service Southeastern Forest Experiment Station. Greber (1983) reported this data in 343 inventory catagories. After grouping the data by species, site class, stocking level, and age class (table 13 in the appendix), 139 stands were recognized (table 14 in appendix). Twenty four different harvest tables were generated using the growth equations provided from a previous study (Greber 1983) (table 15 in the appendix). Stocking levels were considered in growth estimates by making a ratio of actual stand volume to predicted stand volume and multiplying that fraction times the predicted stand volume in the next period. The minimum harvest age was 20 years.

Prices were estimated using stumpage prices from Timber Mart South (1982), $42.67 MBF for hardwood and $78.58 MBF for softwood. Example two held these prices constant for all age classes. Example three assumed a quality premium for larger trees. The prices for softwoods increased linearly from $49.64 at 20 years old, to $78.58 at 80 years old, then remained level. For hardwood prices, increase was linear from $33.17 at 20 years old, to $42.67 at 80 years old, then prices remained level (table 15 in appendix). This price increase was an assumption made to see how the various methods would react to a quality premium.

Since prices were for MBF, volume was converted from cubic foot volume to MBF using the conversion factors of 201.462 cubic feet per thousand board feet for softwood and 225.017 cubic feet per thousand board feet for hardwood (Greber 1983). After the first harvest, all softwood stands were assumed to be planted to 100 percent stocking (stocking #3), and hardwood stands remained at their same stocking level they had before the first harvest.

Results

Tables two and three summarize the results. For example one, binary search found a sustained harvest level of 36,479 MBF over the seven periods with a total present net worth of $2,683,707. Running time was only ten seconds, demonstrating one of the advantages of binary search.

The linear programming run made with MUSYC for non-declining yield found an optimal solution of 38,612 MBF per period with a total present net worth of $2,950,298. The solution time was seven minutes, not including the time MUSYC takes to generate the matrix for LINDO to solve.

For the same example, shadow price search found a solution after 87 iterations that took four minutes to run. The total present net worth was $2,899,276. This solution is $215,569 higher than the binary search solution and $51,022 less than the linear programming solution.

Since the binary search algorithm does not consider timber prices it is not surprising that the present net worth is lower than the linear programming solution. The volume is also less, even though binary search is trying to maximize the harvest flow. An oldest first priority does not consider the growth rate of future stands. If

Table 2 : Result Summary

Example #1 (data set 1)

	Binary Search	Linear programming	Shadow Price Search
Av. Vol. (Mbf/period)	36,479	38,612	39,037
Tot. PNW ($)	2,684	2,950	2,899
Solution Time	10 Sec.	7 Min.	4 Min.

Example #2 (data set 2)

	Binary Search	Linear programming	Shadow Price Search
Av. Vol. (MMBF/period)	12,142	13,117	13,072
Tot. PNW (Million $)	1,494	1,772	1,788
Solution Time	30 Sec.	21 Min.	11 Min.

Example #3 (data set 3)

	Binary Search	Linear programming	Shadow Price Search
Av. Vol. (MMBF/period)	12,142	13,133	13,102
Tot. PNW (Million $)	1,376	1,515	1,534
Solution Time	30 Sec.	20 Min.	10 Min.

Table 3 : Harvest Volume Summary

Example #1 Volumes (MMBF)

	periods						
	1	2	3	4	5	6	7
Binary Search	36.5	36.5	36.5	36.5	36.5	36.5	36.5
L. P.	38.6	38.6	38.6	38.6	38.6	38.6	38.6
S.P.	36.6	41.1	39.8	41.7	40.1	37.6	36.3

Example #2 Volumes (BBF)

	periods						
	1	2	3	4	5	6	7
Binary Search	12.1	12.1	12.1	12.1	12.1	12.1	12.1
L. P.	13.1	13.1	13.1	13.1	13.1	13.1	13.1
S.P.	13.2	14.3	13.2	12.6	12.3	13.7	12.4

Example #3 Volumes (BBF)

	periods						
	1	2	3	4	5	6	7
Binary Search	12.1	12.1	12.1	12.1	12.1	12.1	12.1
L. P.	13.1	13.1	13.1	13.1	13.1	13.1	13.1
S.P.	13.9	13.4	13.0	12.4	12.4	14.2	12.2

more volume can be produced by cutting relatively young stands on high sites early, before older stands on low sites, binary search would not be able to take advantage of this.

Different binary search harvest priority rules could be used such as slowest value growth first or greatest potential future growth first, and these may result in a higher present net worth. But, no one rule can consider all the factors of stand growth and value. A forest is made up of many differing stands, and it is unlikely that one harvest priority rule would produce an optimum result among all the stands.

Examples two and three had 139 stands and a minimum harvest age of 20, so not only were there 44 more stands, each stand had more harvest options. Example one had only 18 harvest options for each stand since the minimum harvest age was 30 and there was a seven period planning horizon. Decreasing the minimum harvest age to 20 causes examples two and three to have 33 harvest options for each stand, resulting in a larger problem.

Binary search found a harvest level of 12,141,720 MBF for both examples two and three, as the only difference between the examples was three recognized a quality premium for larger diameter trees. Since an oldest first harvest priority does not consider value, examples two and three are the same problem for this binary search procedure. Solution time was 30 seconds, much faster than either of the other two methods.

For example two, linear programming found a harvest level of 13,117,078 MBF, and a present net worth was 1.77 billion dollars, higher than the 1.49 billion found by binary search. Solution time was 21 minutes.

Shadow price search took 90 iteration to find a solution within the ten percent deviation for example two. The convergence time was 11 minutes, with a present net worth of 1.79 billion dollars, a little higher than the linear programming solution.

It is interesting that one of the shadow price search solutions found a higher present net worth than the optimal linear programing solution. This occurred because shadow price search only approximately meets the harvest goals, a 10 percent deviation in this case. This flexibility allows the shadow price search solution values to be slightly higher or lower than the optimal linear programming solution values.

Example three produced similar results to example two. Linear programming had a slightly lower PNW of 1.51 billion dollars to shadow price search's value of 1.53 billion dollars. Solution time was 10 minutes, with 79 iterations, for shadow price search, compared to 20 minutes for linear programming.

The present net value for the binary search solution in example three, 1.38 billion dollars, was a little closer to the linear programming and shadow price search present net worths because the value premiums decreased the relative value of the young timber thereby deceasing the profit made by harvesting young stands. This made oldest timber first a better harvest rule.

Comparing the acres harvested by age class per period (tables 4, 5 and 6) clearly shows the close correspondence between shadow price search and linear programming solutions. In all three examples the harvest patterns were very similar. Binary search harvested the older stands first, as instructed, while the other two techniques harvested a variety of stand ages. In example two and three the softwood stands were harvested young and kept on short rotations. This takes advantage of two facts, the softwood growth rates slow down quickly, and regenerated stands are planted to be fully stocked. Converting the low stocked stands to fully stocked stands as soon as possible increases volume and value return.

If the harvest levels from the linear programming solutions are used as the harvest goals in the shadow price search runs convergences are much faster, about half as long, but since the idea is to replace linear programming, the linear programming solution harvest levels would not be known.

The shadow prices produced in the linear programming

Table 4 : Example #1 : Acres Harvested By Age Class Per Period

Linear programming

age	1	2	3	4	5	6	7
30	141	30	150	972	212	883	1,451
40			1	33		647	286
50				95			
60	73	51	159		35	58	
70	94		102	43	103	88	47
80	57		68	148			
90	38			61	35		
100	74	103	153	73	127	53	
110	30	68	72		69		
120	59	55	80		53		
130	80	99	30				
140	47	46			4		
150	42	94	61	30	88		
160	31	85	77		30		
170	47	104	69		40		
180	78	79	62	45			
190	91	45					
200							
210			106	57	98		
220				42	36		

Shadow Price Search

age	1	2	3	4	5	6	7
30	175	30	183	1,159	312	1,144	1,807
40	159		96	105		647	
50	33			123			
60	134	51		69			
70	114	68	69	191			
80	56		73	30			
90	38	57			58		
100	74	103	97	143	54		
110	30	68	72		69		
120		55	30		53		
130	45	158	74				
140	47	94			54		
150	42	30	48	30	44		
160	31	85	41		30		
170	47	76	69		40		
180	43	49	90		100		
190	91	35	30		45		
200			45				
210			127	57	98		
220					36		

Example #1 : Acres Harvested By Age Class Per Period

Binary Search

age	Periods 1	2	3	4	5	6	7
30						195	1,006
40					210	765	648
50					288	507	
60					249		
70					244		
80				112	350		
90				329	129		
100				226			
110			137	263			
120			226	76			
130			203				
140		169	218				
150		218	59				
160	59	241					
170	224	138					
180	216						
190	218						

Table 5: Example #2 : Acres (in thousands) Harvested By Age Class Per Period

Linear programming

age	Periods 1	2	3	4	5	6	7
20	589	270	618	564	2,512	1,549	4,015
30	473	8	606	1,667	1,117		2,229
40	456					674	
50	321					548	
60	241				24	703	
70	128	771	1,068	772	754	601	
80	357	754	215	212	170	117	
90	268	197					
100		359					

Shadow Price Search

age	Periods 1	2	3	4	5	6	7
20	589	268	989	407	2,317	1,344	4,734
30	584	8	608	1,667	1,472	554	1,667
40	456	7				542	
50	321					339	
60	242			5	73	705	
70	128	881	1,168	820	750	553	
80	215	754	106	114	123		
90	462	338					
100		165					

Binary Search

age	Periods 1	2	3	4	5	6	7
20							3,172
30						1,726	2,656
40					1,241	1,897	295
50				862	1,582	682	
60		529	949	1,345	360		
70	742	1,228	1,072	449			
80	554	139					
90	627						

Table 6 : Example #3 : Acres (in thousands) Harvested By Age Class Per Period

Linear programming

age	Periods 1	2	3	4	5	6	7
20	147	110	440	406	2,244	1,220	4,728
30	427	392	443	1,356	1,620	230	1,840
40	456	46	58	323		483	
50	321		117			635	
60	242			42	28	705	
70	128	738	1,168	820	712	597	
80	357	754	248	114	123		
90	355	197					
100		272					

Shadow Price Search

age	Periods 1	2	3	4	5	6	7
20	530	268	543	622	1,988	1,556	4,369
30	427	60	608	1,633	1,447	554	1,633
40	451	52	7			363	
50	321	4	110			692	
60	242				58	705	
70	128	881	1,168	820	754	567	
80	369	754	106	114	123		
90	399	185					
100		228					

Binary Search

age	Periods 1	2	3	4	5	6	7
20							3,172
30						1,726	2,656
40					1,241	1,897	295
50				862	1,582	682	
60		529	949	1,347	360		
70	742	1,228	1,072	449			
80	554	139					
90	627						

run as part of the solution do not necessarily correspond to the shadow prices found in the shadow price search algorithm (table 16 in the appendix). Shadow price search only finds integer solutions, either a stand is harvested or it is not, while linear programming can split stands and harvest portions over several periods. This means with shadow price search there is a range of values over which the shadow prices can vary without changing the solution. Also, if all the shadow prices from a shadow price search solution were multiplied by the same value, the solution of the shadow price search algorithm would not change. As long as the proportional relationships of the shadow prices are not altered, the optimal solutions for the subproblems remain the same.

Conclusions

From this study, it appears that the shadow price search algorithm can be a viable alternative to binary search for approximating linear programming solutions. The running time is longer than binary search, but still less than linear programming, about fifty percent less in our examples. For all the examples, shadow price search found a higher present net worth than binary search. If an integer solution was desired, the solution time savings compared to integer programming would be considerably more. For large scale problems the LP method of branch and bound for integer solutions becomes virtually impossible to solve.

The shadow prices also provide some insight into the cost of meeting the harvest constraints that binary search does not provide. Negative shadow prices indicate the harvest goal is higher than would be attained without the constraint, and positive shadow prices indicate the harvest is less than would be attained without the constraint. It must be remembered, as was previously stated, that the shadow prices only generally correspond to the shadow prices that would be produced by a linear program, so only limited conclusions can be drawn from their values.

One problem with this technique is that there is no guaranteed convergence on a solution within the acceptable limits. The shadow price adjustment procedures are arbitrary, with no mathematical basis for convergence. Whether the solution does converge is dependent on the harvest goals being close to an integer solution, and on the initial adjustment factor value used for the "rough-price" procedure. As an example, using 13,126,028 MBF for the harvest goal with an initial adjustment factor of ten for example three converged within ten percent deviation. For the same problem, using 13,000,000 MBF as a harvest goal trying initial adjustment factors of ten, 14, and 15 only resulted in convergences of 11 to 14 percent. But, a harvest goal of 13,000,000 MBF did converge within the desired limits when an initial adjustment factor of 12 was used. These examples were run only for 100 iterations, it is possible that more iteration would have resulted in closer convergences, but it does point out the problem of not having guaranteed convergence.

In this study harvest goals close to the optimal level were found by starting with the binary search harvest levels and taking the average harvest level after 50 iterations, then rerunning shadow price search with the average harvest level as the goal. Using this type of procedure may be a way of overcoming some of the convergence problems.

Shadow price search works best when there is a large number of stands with relatively small percentages of the total volume in each stand. Also, each stand should have differing value growths so they all behave differently to shadow price variations. If all stands have similar ages and growths, they would tend to be harvested at the same time when using shadow price search, resulting in a "lumpy" harvest schedule. Having a small percentage of the total volume in each stand allows for finer adjustments to be made in harvest levels, since only integer solutions can be found.

Binary search is very limited as it can only handle harvest flow constraints. Shadow price search is a little more versatile. It can handle only one linking constraint, but the constraint need not be harvest flow, as was used in our examples. Any constraint, such as a minimum number of uncut acres left each period over a certain age, or minimum browse production for wildlife, could be used. The only restriction is that the subproblems must be totally separable, meaning they can be optimized independently and the production summed over all stands to find the total. A wildlife or sediment constraint that was dependent on how many stands in a particular drainage were harvested could not be used, nor could adjacency constraints be applied.

In summary, the limitations of shadow price search in comparison to binary search are:

1) Slower,
2) Harder to use.

It's limitation in comparison to linear programming are:

1) Does not handle constraints beyond one set of linking constraints for the subproblems.

Limitations relative to both methods are:

1) No guaranteed convergence on an acceptable solution,
2) Unable to reach harvest goals exactly,
3) Needs many stands with a relatively small portion of the total volume in each stand.

The advantages of shadow price search relative to binary search are:

1) Capable of more closely approximating an optimal LP solution,
2) Produces shadow prices along with the solution, providing some insight into the cost of meeting harvest goals,
3) Finds a solution that is optimal for the actual harvest levels that result.

It's advantages relative to linear programming are:

1) Faster to formulate and solve
2) Gives an integer solution (if one is desired).

Shadow price search is promising but still somewhat experimental. Perhaps it's most valuable use is it's ability to find integer solutions. There is room for further development and improvements, especially in the shadow price adjustment procedures (see problems and future research).

Problems And Future Research

The shadow price adjustment procedures developed here are not the only ones that could be used. It is likely that more efficient heuristic methods could be found. "Newprice" occasionally computes a very large change in a shadow price. When the computed change is greater than 50 percent of the original shadow price the "defaultprice" procedure was used instead in this study.

Using previous knowledge about the harvest scheduling problem being solved could assist in the solution process. Knowing the general pattern of the shadow prices over the planning horizon from previous LP or shadow price search solutions and using these prices for a starting estimate would reduce the number of iteration required. Also, general knowledge about the forest's available volume over time could be used to give a more reasonable starting guess than all zeros, which was the starting estimate used in this study for the shadow prices. If there is a lack of volume in the middle planning periods, the shadow prices could be set positive for the early periods, drop to negative values in the middle periods, then rise to become positive for the final periods.

Standard methods for "Lagrange Relaxation", such as subgradients, gradient and tangential approximation, do have a mathematical basis for convergence on a solution and they are able to handle more than one linking (complicating) constraint. They also involve more mathematical computation, as was mentioned earlier. How these methods work in practice on harvest scheduling problems and what the solution times actually are needs to be more fully explored.

It is likely that more efficient computer code could be written that would reduce the computational burden and shorten the running time. Also, the program should be expanded to allow the user to specify a level of inventory left at the end of the planning horizon.

References

Bazaraa, M.S., and Sherali. 1981. On the choice of step size in subgradient optimization. European Journal Of Operations Research, pp. 380-388

Berck, P., and Bible, T.. 1984. Solving and interpreting large-scale harvest scheduling problems by duality and decomposition. Forest Science, Vol.30, pp. 173-184.

Brooks, R.B.S., and Geoffrion A.M.. 1966. Finding Everett's multipliers by linear programming. Operations Research 14, pp.1149-1152.

Clutter, J.L., and others. 1968. MAX MILLION-- A computerized forest management planning system. Sch Forest Resour, Univ. GA, Athens, 61p.

Dantzig, G. B., and Wolf, P.. 1961. The Decomposition algorithms for linear programs. Econometrica, Vol. 29, No. 767, pp.767-778.

Everett, H.. 1963. Generalized Lagrange multiplier method for solving problems of optimum allocation of resources. Operations Research (11), pp.399-417.

Geoffrion. A.M.. 1974. Lagrangean relaxations for integer programming. mathematical programming study 2. pp. 82-114

Greber, Brian J.. 1983. Development of a joint product timber supply model. Dissertation. Virginia PolyTechnic insitute and State University, Blacksburg, Virginia pp.325

Held, M., Wolf, P. and Crowder, H.D.. 1974. Validation of subgradient optimization. mathematical programming (6), pp.62-88

Hoganson, Howard M. and Dietmar W. Rose. 1984. A simulation approach for optimal Timber Management Scheduling. Forest Science vol. 30, pp.220-238

Hoganson, H. M.. 1981. Timber management scheduling under uncertainty: two alternative approaches. Dissertation, University of Minnesota pp. 150

Johnson, K.N., H.L. Scheurman, and John Beuter. 1975. Oregon timber resource model-- a brief introduction. In Systems Analysis and Forest Resource Management (J. Meadows and others, eds.). pp. 148-163 Soc Am For. 457 p.

Johnson, K. N. and H. Scheurman. 1977. Techniques for prescribing optimal timber harvest and investment under different objectives--discussion and synthesis. Forest Science Monograph 18, pp.31

Johnson, K. N. and D. B. Jones. 1979. a users guide to multiple use-sustained yield resource scheduling calculation (MUSYC). Timber Management. USDA Forest Service, Fort Collins, CO., 242 p. mimeo.

Johnson, K. N. and Philip L. Tedder. 1983. Linear programming vs. binary search in periodic harvest level calculation. Forest Science, vol. 29 (3). pp. 569-581.

Johnson, K. N., Tom Stuart, and Sarah Crim. 1986. FORPLAN version 2-- an overview. Land Management Planning. USDA Forest Service, Fort Collins, CO.

Lasdon, L.S.. 1972. Decomposition in resource allocation. Paper from 1972 Conference NATO Ist.. Cambridge, England . Decomposition OF Large Scale Problems. American Elsevier Pub. Co. Inc., New York, 1973, Editor Himmelblau, D.M.. pp. 207-231

Lasdon, L.S.. 1970. Optimization theory for large systems. Macmillian, London, p.523

Littschwager, J.M., and Tcheng, T.H.. 1967. Solution of a large-scale forest scheduling problem by LP decomposition. Journal of Forestry 65, pp. 644-646

Murty, K.G., 1976. Linear and combinatorial programming. Wiley, New York, pp.555

Navon, D.I.. 1971. Timber RAM...A long-range planning method for commercial timber lands under multiple use management.. USDA Forest Service Res. Pap PSW-70, 22 p. Pac Southwest Forest and Range Exp Stn, Berkely.

Nazareth, L.. 1980. A land management model using Dantzig-Wolf decomposition. Management Science, Vol. 26, No. 5, May , pp. 510-523.

Perry, C.,Crellin, K.C.. 1982. The precise meaning of a shadow price. Interfaces, Vol. 12 No.2 April. pp. 61-62

Sassaman, R,W., E. Hold, and K. Bergsvick. 1972 User's manual for a computer program for simulating intensively managed allowable cut. U.S. Forest Service General Technical Report PNW-1.

Tedder, P. L., J. S. Schmidt, and Gourley. 1980. TREES: Timber resource economic estimation system. Volume I-a User's Guide For Forest Management And Harvest Scheduling.

Forest res. Lab. Oregon State Univers., Res. Bull. 31a, p.81

Walker, J.L.. 1976. ECHO: Solution technique for a nonlinear economic harvest optimization model. In Systems Analysis And Forest Resource Management (J. Meadows, B. Bare K. Ware, and C. Row, eds.). Soc Am For, Wash, D.C. 457 p.

APPENDICES

Table 7 : Example #1 Data Inventory

Type #1			Type #2		
Stand #	intial age	acres	Stand #	initial age	acres
1	10	47	20	10	58
2	20	88	21	20	35
3	30	34	22	30	35
4	40	79	23	40	69
5	50	30	24	50	58
6	60	73	25	60	54
7	70	69	26	70	43
8	80	53	27	80	43
9	90	42	28	90	46
10	100	54	29	100	68
11	110	44	30	110	55
12	120	30	31	120	58
13	130	40	32	130	46
14	140	36	33	140	64
15	150	69	34	150	45
16	160	60	35	160	30
17	170	49	36	170	49
18	180	36	37	180	35
19	190	36	38	190	46

Type #3			Type #4		
Stand #	intial age	acres	Stand #	initial age	acres
39	10	33	58	10	92
40	20	66	59	20	30
41	30	64	60	30	77
42	40	71	61	40	88
43	50	51	62	50	33
44	60	73	63	60	61
45	70	54	64	70	30
46	80	57	65	80	54
47	90	38	66	90	30
48	100	44	67	100	30
49	110	30	68	110	44
50	120	41	69	120	30
51	130	45	70	130	48
52	140	47	71	140	30
53	150	42	72	150	51
54	160	31	73	160	30
55	170	47	74	170	49
56	180	43	75	180	57
57	190	45	76	190	49

(Table 7 Continued)

Example #1 Data Inventory

Type #5

Stand #	intial age	acres
77	10	58
78	20	30
79	30	34
80	40	43
81	50	69
82	60	68
83	70	30
84	80	56
85	90	57
86	100	30
87	110	30
88	120	59
89	130	48
90	140	41
91	150	34
92	160	46
93	170	30
94	180	45
95	190	42

Source : Hypothetical data set

Table 8 : Example #1 Volume And Economic data

Type #1

	Initial		Regenerated	
age (years)	volume (MBF/acre)	Net. Value ($/MBF)	volume (MBF/acre)	Net. Value ($/MBF)
30	5	20	16	15
40	10	20	20	15
50	15	20	24	15
60	20	23	28	30
70	25	23	32	30
80	29	23	33	30
90	32	23	34	36
100	34	27	35	36
110	38	27	--	--
120	40	27	--	--
130	42	27	--	--
140	44	27	--	--
150	46	27	--	--
160	48	29	--	--
170	50	29	--	--
180	51	29	--	--
190	52	29	--	--
200	53	29	--	--
210	54	31	--	--
220	55	31	--	--
230	56	31	--	--
240	57	31	--	--
250	58	31	--	--
260	59	31	--	--

Source : Hypothetical data

Table 9 : Example #1 Volume And Economic data

Type #2

	Initial		Regenerated	
age (years)	volume (MBF/acre)	Net. Value ($/MBF)	volume (MBF/acre)	Net. Value ($/MBF)
30	12	15	19	12
40	17	15	23	12
50	22	15	27	12
60	27	25	30	23
70	32	25	33	23
80	36	25	36	23
90	40	25	39	26
100	43	35	41	26
110	45	35	--	--
120	47	35	--	--
130	49	35	--	--
140	51	35	--	--
150	53	35	--	--
160	55	35	--	--
170	57	35	--	--
180	58	45	--	--
190	60	45	--	--
200	61	45	--	--
210	62	45	--	--
220	63	47	--	--
230	64	47	--	--
240	65	47	--	--
250	66	47	--	--
260	67	47	--	--

Source : Hypothetical data

Table 10 : Example #1 Volume And Economic data

Type #3

	Initial		Regenerated	
age (years)	volume (MBF/acre)	Net. Value ($/MBF)	volume (MBF/acre)	Net. Value ($/MBF)
30	12	30	23	28
40	18	30	29	28
50	24	30	35	28
60	28	34	40	40
70	32	34	43	40
80	36	34	46	40
90	40	34	49	47
100	42	34	51	47
110	44	34	--	--
120	46	34	--	--
130	48	42	--	--
140	50	42	--	--
150	52	42	--	--
160	53	42	--	--
170	54	42	--	--
180	55	42	--	--
190	56	42	--	--
200	56	42	--	--
210	56	42	--	--
220	56	42	--	--
230	56	42	--	--
240	56	42	--	--
250	56	42	--	--
260	56	42	--	--

Source : Hypothetical data

Table 11 :Example #1 Volume And Economic data

Type #4

	Initial		Regenerated	
age (years)	volume (MBF/acre)	Net. Value ($/MBF)	volume (MBF/acre)	Net. Value ($/MBF)
30	6	10	16	25
40	9	10	20	25
50	12	10	23	25
60	14	15	26	25
70	16	15	29	27
80	18	15	33	27
90	20	15	36	27
100	22	16	38	27
110	24	20	--	--
120	26	20	--	--
130	28	20	--	--
140	30	20	--	--
150	32	30	--	--
160	34	30	--	--
170	36	30	--	--
180	38	30	--	--
190	40	30	--	--
200	42	30	--	--
210	44	40	--	--
220	45	40	--	--
230	46	40	--	--
240	47	40	--	--
250	48	40	--	--
260	49	40	--	--

Source : Hypothetical data

Table 12 :Example #1 Volume And Economic data

Type #5

	Initial		Regenerated	
age (years)	volume (MBF/acre)	Net. Value ($/MBF)	volume (MBF/acre)	Net. Value ($/MBF)
30	16	10	25	30
40	20	10	29	30
50	24	10	33	40
60	28	10	37	40
70	32	25	41	40
80	35	25	45	50
90	38	25	49	50
100	40	27	52	50
110	42	27	--	--
120	44	27	--	--
130	46	27	--	--
140	48	27	--	--
150	50	27	--	--
160	52	28	--	--
170	54	28	--	--
180	56	28	--	--
190	58	28	--	--
200	60	28	--	--
210	62	28	--	--
220	64	33	--	--
230	66	33	--	--
240	68	33	--	--
250	31	33	--	--
260	34	33	--	--

Source : Hypothetical data

Table 13 : <u>Stand Data Grouping For Examples 2 And 3</u>

<u>Species</u> : Hardwood

Softwood

<u>Site</u> : High - Yields more than 85 cubic ft./year average at the culmination of mean annual increment for the given cover type.

Medium - Yields 50 to 85 cubic ft./year average at the culmination of mean annual increment for the given cover type.

Low - Yields less than 50 cubic ft./year average at the culmination of mean annual increment for the given cover type.

<u>Stocking</u> : 1 - less than 39%

2 - 40% to 79%

3 - 80% to 119%

4 - Greater than 120%

<u>Initial Age Classes</u> :	0 to 9 years	50 to 59 years
(based on av. age)	10 to 19 years	60 to 69 years
	20 to 29 years	70 to 79 years
	30 to 39 years	80 to 90 years
	40 to 49 years	

Table 14 : Inventory Data For Examples 2 And 3

Stand #	Init. age	acres	Species	Site	Stocking
1	10	3712	Softwood	Low	1
2	20	29412	Softwood	Low	1
3	30	3050	Softwood	Low	1
4	40	2153	Softwood	Low	1
5	10	36275	Softwood	Med.	1
6	20	44540	Softwood	Med.	1
7	40	26411	Softwood	Med.	1
8	10	70147	Softwood	High	1
9	20	47202	Softwood	High	1
10	30	3965	Softwood	High	1
11	40	7100	Softwood	High	1
12	50	3802	Softwood	High	1
13	30	3388	Softwood	low	2
14	60	3873	Softwood	low	2
15	80	4617	Softwood	low	2
16	20	409492	Softwood	Med.	2
17	30	340506	Softwood	Med.	2
18	40	3105	Softwood	Med.	2
19	50	12896	Softwood	Med.	2
20	60	3803	Softwood	Med.	2
21	30	67464	Softwood	High	2
22	30	12264	Softwood	Low	3
23	40	25028	Softwood	Low	3
24	20	7980	Softwood	Med.	3
25	30	33717	Softwood	Med.	3
26	40	291520	Softwood	Med.	3
27	50	10707	Softwood	Med.	3
28	60	7111	Softwood	Med.	3
29	70	90042	Softwood	Med.	3
30	80	4192	Softwood	Med.	3
31	30	9050	Softwood	High	3
32	40	95790	Softwood	High	3
33	60	3965	Softwood	High	3
34	10	52835	Softwood	Low	4
35	40	4440	Softwood	Low	4
36	50	23598	Softwood	Low	4
37	70	2889	Softwood	Low	4
38	80	4534	Softwood	Low	4
39	10	553542	Softwood	Med.	4
40	50	182693	Softwood	Med.	4
41	60	151577	Softwood	Med.	4
42	70	13051	Softwood	Med.	4
43	80	21465	Softwood	Med.	4
44	90	20166	Softwood	Med.	4
45	10	1567	Softwood	High	4
46	50	86995	Softwood	High	4
47	60	71377	Softwood	High	4

(Table 14 Continued)

48	70	21605	Softwood	High	4
49	80	13910	Softwood	High	4
50	90	8099	Softwood	High	4
51	10	17128	Hardwood	Low	1
52	20	6843	Hardwood	Low	1
53	30	6592	Hardwood	Low	1
54	40	3802	Hardwood	Low	1
55	60	3730	Hardwood	Low	1
56	70	6664	Hardwood	Low	1
57	10	10638	Hardwood	Med.	1
58	20	14908	Hardwood	Med.	1
59	30	11914	Hardwood	Med.	1
60	40	9754	Hardwood	Med.	1
61	60	3881	Hardwood	Med.	1
62	90	22827	Hardwood	Med.	1
63	10	4576	Hardwood	High	1
64	20	13384	Hardwood	High	1
65	30	4534	Hardwood	High	1
66	50	3881	Hardwood	High	1
67	80	3428	Hardwood	High	1
68	10	68281	Hardwood	Low	2
69	20	51704	Hardwood	Low	2
70	30	25750	Hardwood	Low	2
71	40	3175	Hardwood	Low	2
72	50	7234	Hardwood	Low	2
73	60	3803	Hardwood	Low	2
74	80	7797	Hardwood	Low	2
75	90	8654	Hardwood	Low	2
76	10	16586	Hardwood	Med.	2
77	20	32035	Hardwood	Med.	2
78	30	20140	Hardwood	Med.	2
79	40	18028	Hardwood	Med.	2
80	50	5944	Hardwood	Med.	2
81	60	18292	Hardwood	Med.	2
82	70	8284	Hardwood	Med.	2
83	80	12407	Hardwood	Med.	2
84	90	7983	Hardwood	Med.	2
85	30	16081	Hardwood	High	2
86	40	7867	Hardwood	High	2
87	60	16845	Hardwood	High	2
88	70	3827	Hardwood	High	2
89	80	44790	Hardwood	High	2
90	90	9033	Hardwood	High	2
91	10	72888	Hardwood	Low	3
92	30	84621	Hardwood	Low	3
93	40	80661	Hardwood	Low	3
94	50	53841	Hardwood	Low	3
95	60	59697	Hardwood	Low	3
96	70	12167	Hardwood	Low	3
97	80	20674	Hardwood	Low	3

(Table 14 Continued)

98	90	22869	Hardwood	Low	3
99	10	535140	Hardwood	Med.	3
100	20	460135	Hardwood	Med.	3
101	30	58012	Hardwood	Med.	3
102	40	572656	Hardwood	Med.	3
103	50	887218	Hardwood	Med.	3
104	60	633058	Hardwood	Med.	3
105	70	435478	Hardwood	Med.	3
106	80	256923	Hardwood	Med.	3
107	90	52102	Hardwood	Med.	3
108	10	78158	Hardwood	High	3
109	30	52802	Hardwood	High	3
110	40	87725	Hardwood	High	3
111	50	111938	Hardwood	High	3
112	60	123859	Hardwood	High	3
113	70	61417	Hardwood	High	3
114	80	2671	Hardwood	High	3
115	90	59802	Hardwood	High	3
116	40	34992	Hardwood	Low	4
117	50	52523	Hardwood	Low	4
118	60	38307	Hardwood	Low	4
119	70	126114	Hardwood	Low	4
120	80	87132	Hardwood	Low	4
121	90	113713	Hardwood	Low	4
122	10	56546	Hardwood	Med.	4
123	20	60362	Hardwood	Med.	4
124	30	549210	Hardwood	Med.	4
125	40	74057	Hardwood	Med.	4
126	50	34922	Hardwood	Med.	4
127	60	83016	Hardwood	Med.	4
128	70	96249	Hardwood	Med.	4
129	80	50997	Hardwood	Med.	4
130	90	288302	Hardwood	Med.	4
131	10	44502	Hardwood	High	4
132	20	41575	Hardwood	High	4
133	30	41575	Hardwood	High	4
134	40	49485	Hardwood	High	4
135	50	123690	Hardwood	High	4
136	60	2034	Hardwood	High	4
137	70	3428	Hardwood	High	4
138	80	18180	Hardwood	High	4
139	90	13621	Hardwood	High	4

Source: 1976 Virginia forest inventory conducted by the U.S.Forest Service Southeastern Forest Experiment Station. Reported by Brian Greber 1983.

Table 15 : Volume Data For Examples 2 and 3

Softwood Yield Equations

Site Class

Site Class	
Low	Log(hrdvol)=(5.15076 - 11.91171/age)/225.017 Log(sftvol)=(6.15122 - 20.22186/age)/201.462
Med.	Log(hrdvol)=(5.06919 - 11.09309/age)/225.017 Log(sftvol)=(7.42915 - 23.5669/age)/201.462
High	Log(hrdvol)=(5.4619 - 7.9405/age)/225.17 Log(sftvol)=(7.60753 - 17.2721/age)/201.017

Hardwood Yield Equations

Site Class

Site Class	
Low	Log(hrdvol)=(4.06784 + 0.03127age)/225.017 Log(sftvol)=(4.91902 - 11.65846/age)/201.462
Med.	Log(hrdvol)=(4.18507 + 0.03855age + 0.00061 age^2 - 0.000008age^3)/225.017 Log(sftvol)=(5.14304 - 10.59178/age)201.462
High	Log(hrdvol)=(4.19552 + 0.07857age - 0.00027 age^2 - 0.000002age^3)/225.017 Log(sftvol)=(5.47877 - 4.68816/age)/201.462

totvol=stk(sftvol + hrdvol)

---

hrdvol = MBF/acre of fully stocked hardwood
sftvol = MBF/acre of fully stocked softwood
totvol = total volume at specified stocking level
stk = percent stocking

Economic Data Example 2

Hardwood value($/MBF) = 42.67
Softwood Value($/MBF)= 78.58

Economic Data Example 3

Hardwood value = 30 + age12.67/80	for age <= 80
Hardwood value = 42.67	for age > 80
Softwood value = 40 + age38.58/80	for age <= 80
Softwood value = 78.58	for age > 80

Table 16 : Shadow Prices From The Linear Programming And Shadow Price Search Solutions

Example 1:

	periods						
	1	2	3	4	5	6	7
L.P.	--	-15.96	-22.02	-22.43	-19.91	-12.55	-5.76
S.P.	1.30	-13.96	-21.09	-24.85	-35.23	-29.72	-22.03

Example 2:

	periods						
	1	2	3	4	5	6	7
L.P.	--	-20.92	-28.01	-27.68	-22.20	-13.86	-4.79
S.P.	2.12	-13.58	-18.71	-21.42	-20.47	-19.81	-11.08

Example 3:

	periods						
	1	2	3	4	5	6	7
L.P.	--	-18.27	-23.18	-21.45	-16.32	-10.52	-4.00
S.P.	-0.94	-16.85	-22.16	-24.89	-22.89	-21.71	-13.13