United States
Department of
Agriculture

Forest Service

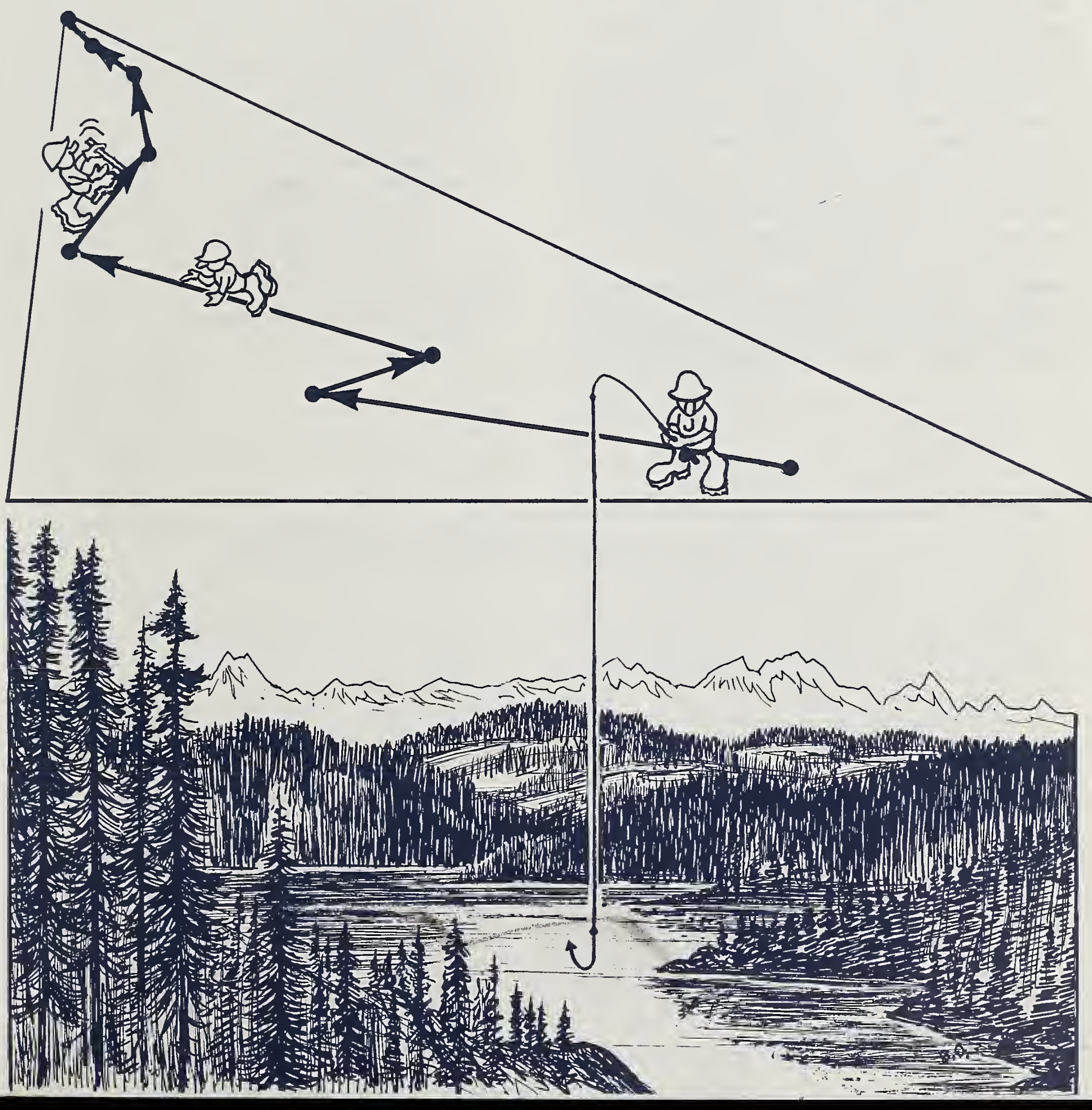
Intermountain
Research Station

General Technical
Report INT-269

June 1990

A Programmer's Guide to the Prognosis Optimization Model

Robert A. Monserud
Robert G. Haight



Support for this research came from the following branches of the Forest Service: Intermountain Research Station, Northern Region of the National Forest System, and the Timber Management Research Staff from the Washington Office. We are indebted to Bill Wykoff and Nick Crookston for invaluable programming assistance.

THE AUTHORS

ROBERT A. MONSERUD is principal mensurationist, Intermountain Research Station, at the Forestry Sciences Laboratory in Moscow, ID. After earning a B.A. degree in mathematics at the University of Iowa in 1968 and teaching high school math for 3 years, Monserud switched to forestry. He earned an M.S. degree in forest management in 1973 and a Ph.D. degree in forest biometrics and mensuration in 1975, both at the University of Wisconsin, Madison. Since joining the Intermountain Research Station in 1975, Monserud has worked on a variety of problems centered on modeling stand dynamics and site productivity for uneven-aged mixed-species forests. For the past 6 years, Monserud has served as associate editor of *Forest Science*. He has also served as invited visiting professor at University of California, Berkeley, and as invited visiting scientist at the International Institute for Applied Systems Analysis in Laxenburg, Austria. Current projects include problems in optimal stand management (conducted with Haight), modeling the components of tree growth, site productivity, and crown backdating, and global vegetation modeling.

ROBERT G. HAIGHT is principal economist, Southeastern Forest Experiment Station at the Forestry Sciences Laboratory in Research Triangle Park, NC. Haight earned a B.S. degree in forestry in 1978 and an M.F. degree in 1981, both at the University of California in Berkeley. In 1985 he earned a Ph.D. degree in forest management at Oregon State University. After spending 2 years as Ciriacy-Wantrup Postdoctoral Fellow at the University of California, Berkeley, Dr. Haight joined the Southeastern Forest Experiment Station in 1987. He has authored several important publications on optimal management of both even-aged and uneven-aged forest stands, as well as a book on population harvesting coauthored with Wayne Getz.

CONTENTS

The use of trade or firm names in this publication is for reader information and does not imply endorsement by the U.S. Department of Agriculture of any product or service.

A Programmer's Guide to the Prognosis Optimization Model

Robert A. Monserud
Robert G. Haight

INTRODUCTION

The Prognosis Optimization Model is designed as a research tool to investigate the effects of changes in economic and biological parameters on optimal harvest regimes for mixed-species stands in the Northern Rocky Mountains. The optimization program involves a coordinate-search process called the Method of Hooke and Jeeves (Hooke and Jeeves 1961), and it incorporates without modification version 5.2 of the Stand Prognosis Model (Wykoff et al. 1982; Ferguson and Crookston 1984; Crookston 1985; Ferguson et al. 1986; Hamilton 1986; Wykoff 1986). The optimizer requires the same keyword and treelist input files as the Prognosis Model, as well as an additional input file that defines the size and objective of the harvest problem. The user sets the size of the problem by defining harvest classes that consist of diameter classes, species groups, and periods in which harvesting may take place. The management objective may either be to maximize present value or to maximize volume production. The program returns optimal harvest intensities for the predefined harvest classes. Because of their compatibility, results from the Prognosis Optimization Model complement simulation results obtained with the Stand Prognosis Model.

This guide describes how to compile and run the optimization program using a personal computer. In section 2 we describe the kinds of problems that may be solved and some limitations. Section 3 describes the input requirements and output associated with the optimization model. The input and output for three optimization runs are described in section 4. Section 5 describes how to organize the input files and execute the optimizer. Section 6 describes how to install the optimizer on a personal computer. The file names used in sections 5 and 6 are identical to those contained in the distribution diskette, which may be obtained from the senior author. Readers interested in using the existing executable module for optimization work should read the first six sections. Those interested in changing and compiling the source code should read sections 7 and 8, which describe in detail the source code and compilation instructions.

There are other sources of information related to the Prognosis Optimization Model. We assume that readers are familiar with the operation of the Stand Prognosis Model (see Wykoff et al. 1982; Wykoff 1986; Ferguson and Crookston 1984; Crookston 1985; Ferguson et al. 1986; Hamilton 1986). Those interested in changing the program should also read Wykoff's (1988) instructions for compiling the Prognosis Model on a personal computer. The performance of the Prognosis Optimization Model and its application to various management problems are described by Haight and Monserud (1990a, b).

The following conventions are used to describe programs, variables, and files. Fortran PROGRAM NAMES are written in capital letters; *VARIABLES* in the programs are written in capital italics. Except where noted, Fortran programs are found in MS-DOS files with the same names. When DOS files are described, they are written in capital italics with a suffix *(MAIN.FOR)* Prognosis KEYWORDS are written in small capitals, and they will always be identified as keywords in the text.

THE PROGNOSIS OPTIMIZATION MODEL

The Stand Prognosis Model is designed to simulate the development of forest stands in the Northern Rocky Mountains. The individual tree is the basic unit of projection, and stands with any combination of species and size classes can be accommodated. We use the Inland Empire version 5.2 of the Prognosis Model, which is calibrated for 11 conifer species on 30 habitat types in northern Idaho and northwestern Montana.

The Prognosis Optimization Model is capable of solving harvesting problems that fit two broad stand-management categories: any-aged management and even-aged management. The any-aged stand management problem is to determine the best sequence of species and diameter-class harvesting rates and planting intensities for an existing stand over an infinite time horizon. Harvesting and planting may take place simultaneously in any period, and there are no constraints on the stand size or age structure. Because the sequence of residual diameter distributions may take on any structure, the optimal management regime is any-aged (see Haight and Monserud 1990a for a mathematical description of the problem). In practical applications, an infinite-time-horizon problem cannot be solved. However, sensitivity analysis can be used to determine the impacts of finite-time-horizon approximations. Haight and Monserud (1990a) formulate any-aged management problems in which the stand is eventually clearcut. They find that, with a maximum present value objective and a 4 percent discount rate, horizons greater than 160 years have little impact on optimal management regimes and present values. Haight and Monserud (1990b) determine the impacts of maximum present value and maximum cubic foot volume objectives on optimal any-aged management regimes.

For even-aged management, harvesting problems have two components: the determination of the best thinning regime to employ in an existing stand before it is clearcut (or seedtree harvested), and the determination of the best sequence of thinnings to undertake in the subsequent plantation (or even-aged natural stand). These components are called the conversion problem and the plantation problem, respectively. The optimal plantation regime is found independently of the conversion regime, and it maximizes an infinite series of rents. (Rent is a general term for payoffs that may be measured in present value or volume units depending on the problem objective. Haight and Monserud [1990b] give detailed solutions to even-aged management problems that are formulated with these two objectives.) The best infinite-series plantation is found by comparing the rents from plantations with different rotation ages. The optimal conversion regime maximizes the sum of the rents from conversion and plantation management. The optimal conversion period is found by comparing the rents from conversion and plantation regimes with different conversion ages.

A third category of harvesting problems involves uneven-aged stand management. The goal is to determine the best sequence of selection harvests for a transition regime that terminates with a steady state. The development of methods for solving these constrained problems is in the planning stage, and they are not included in this version of the optimizer.

The coordinate-search algorithm does not guarantee convergence to a globally optimal solution to a given harvesting problem. For both even-aged and any-aged management problems, the Prognosis Optimization Model evaluates harvest regimes that are defined by a set of control variables representing the fractions of trees harvested by diameter class and species group in specified periods. The number of diameter classes and species groups greatly affects convergence. Performance tests show that, for problems with diameter-class and species-class harvest controls defined in two or more periods, the algorithm converges to solutions that may vary considerably in present value and harvest pattern depending on the point at which the search starts and how broadly the control variables are defined (Haight and Monserud 1990a).

From our experience, the performance of the optimizer is improved by defining harvest controls for relatively wide diameter classes and broad species groupings. In the examples presented here, we use three submerchantable diameter classes (0-2, 2-4, and 4-7 inches) and five merchantable diameter classes (7-10, 10-14, 14-18, 18-22, and 22-40 inches). These groupings have proven efficient and considerably reduce the variability resulting from different random starts (Haight and Monserud 1990a). We recommend using species groups in addition to diameter classes only when stumpage price varies by species group. For maximum volume production objectives in which stumpage price is not a concern, the optimizer usually converges to solutions with higher objective function values using just one species group.

Because the optimizer does not guarantee the global optimum, we recommend examining a range of local optima before selecting the best solution to a given problem. In our management studies, we solve each harvesting problem at least three times (and usually six) using different starting points for the harvest controls. Starting points are determined either randomly or from solutions to problems with shorter time horizons.

While the coordinate-search algorithm concentrates on determining optimal harvest intensities by diameter class and species group, planting and site preparation may also be scheduled in any period. These cultural activities are defined as fixed rather than variable inputs. The coordinate-search process determines the optimal harvest proportions for a given set of cultural activities. Repeated application of the optimization program allows comparison of the efficiencies of sets of cultural activities. In an example presented in section 4.3, the Prognosis Optimization Model is used to determine the thinning regime that maximizes merchantable cubic foot volume production for a white pine plantation that starts with 600 seedlings per acre and has a rotation age of 120 years. To determine the best planting density, the optimization needs to be repeated for a wide range of planting densities (see, for example, Haight and Monserud 1990b). Performance tests showed that this procedure provides solutions with higher objective function values than does a procedure in which cultural activities are treated as control variables in addition to the harvest controls (Haight and Monserud 1990a).

3 INPUT REQUIREMENTS AND OUTPUT PRODUCED

3.1 Optimizer Input File

The input file for the optimizer contains records. Each record is a line containing one or more numerical entries. Unless otherwise specified, each entry is assigned a five-space column. Each entry is either an integer, which is right justified within its column, or a real number, which includes a decimal point and may be placed anywhere within its column. For an example of the optimizer input, see figure 1a, section 4.1. Those interested in the Fortran code that defines the input format may refer to *MAIN.FOR* in the appendix. In the following description, variable names are taken directly from the Fortran code.

The first entry in record 1 is an integer (*NRUN*) representing the number of separate optimizations to be made during the execution of the program. Then, on the same record are *NRUN* entries representing random number seeds. These are coded as integer-valued real numbers. Each optimization run uses a different seed. If a seed is positive, the optimization will begin by using a set of random harvest controls generated by that seed. If the seed is negative, the optimization will start by using specified harvest controls (see record type 8, below). This convention allows for combining optimizations that begin with different random seeds along with an optimization that begins with specific harvest controls, all in the same batch job.

Record 2 has five real-number entries that define the discount rate and parameters for the search procedure (see Haight and Monserud 1990a):

1. *R* is the decimal-valued discount rate (between 0.0 and 1.0).
2. *ALPHA* is the acceleration step used in the pattern search.
3. *DELTA* is the beginning step size for the coordinate search.
4. *EPS* is the minimum step size allowed in the coordinate search before termination.
5. *EPS1* is the minimum improvement in present value resulting from successive coordinate searches allowed before termination.

The discount rate is used to evaluate harvest revenues. When $0.0 < R \leq 1.0$, the objective is to maximize present value. When $R = 0.0$, a maximum volume yield objective is implied. Based on our experience, we suggest using the following values for *ALPHA*, *DELTA*, *EPS*, and *EPS1*. Setting *ALPHA* = 0.1 allows for a moderate change in the harvest control values each time a pattern search is performed. Setting *DELTA = 1.0* tests the boundary of each control during the first coordinate search. Setting *EPS = 0.2* and *EPS1 = 999.0* allows five coordinate-search iterations before termination.

Record 3 contains seven integer entries that define the size of the optimization problem and the features of the output:

1. *NUMCYC* is the number of growth periods in the planning horizon. (This has the same value as the NUMCYCLE keyword in Prognosis.)
2. *MERCH* is the index of the minimum merchantable diameter class (for example, if the first three diameter classes contain unmerchantable trees, *MERCH = 4*).
3. *NGROUP* is the number of species groups used to classify trees.
4. *MVOL* is an index for the unit of volume measurement used in evaluating management regimes. When *MVOL* > 0, merchantable board foot volume is used. When *MVOL* < 0, merchantable cubic foot volume is used.
5. *LENGTH* is the length of the growth period (usually 10 years).
6. *NTH* specifies that output be printed every *NTH* growth period.
7. *NOKEY* is zero if the THINDBH keywords describing the optimal regime are not to be written on the optimizer output file.

All entries on record 4 are integers. The first entry is the number of harvest periods *NCUTS*. The remaining entries are the periods (vector *ICUT*) in which harvests are scheduled. Note that the start of period 1 in the optimizer corresponds to year 0 in Prognosis and therefore denotes an immediate harvest. Also note that a clearcut in the final period always takes place and thus is not a variable in the optimization.

The first entry on record 5 is an integer representing the number of diameter-class boundaries (*NCLASS* + 1). Following this entry are *NCLASS* + 1 real numbers representing the boundaries. The first boundary is 0.0, and the last should be larger than any tree you expect the optimizer to see. The optimizer then calculates class midpoints, which are used in the output routine to summarize the diameter distribution. The diameter classes defined by these boundaries classify tree records in Prognosis for the purpose of harvesting. A harvest control variable is defined for each diameter class.

The next set of records (call these record type 6) enters the species codes for the species groups. There is one record for each of the *NGROUP* species groups. The first entry is an integer representing the number of species. The following entries are real numbers representing species identification codes that Prognosis uses (see Wykoff 1986). As a shortcut when all species are assigned to one group (*NGROUP = 1*), simply enter 1 for the number of species in the group and 0.0 as the species code (see fig. 1a, section 4.1).

Prices for each species group are entered with the next set of records (record type 7). Again, there is one record for each species group, and these records are in the same order as the species-group records written above. The entries on each record are real numbers representing prices by diameter class. The prices for unmerchantable diameter classes are negative and represent the cost per tree for precommercial thinning. The prices for merchantable diameter classes represent stumpage prices per thousand board feet or thousand cubic feet, depending on the value of *MVOL*. If merchantable volume maximization is the objective, the prices for the unmerchantable and merchantable classes should be 0.0 and 1.0, respectively (the discount rate *R* should also be 0.0).

The final sets of records (record type 8) enter the initial values for the harvest control array *U1*. First of all, there is one set of records for each species group. The record sets are in the same order as the species-group records written above. There are *NUMCYC* records for each species group. Each of these records contain the harvest control values by diameter class for each period, beginning in the first period. Note that harvests occur at the start of a period and that all optimizations end with a clearcut that capitalizes the growing stock. The entries in each record are real numbers between 0.0 and 1.0 that represent the harvest proportions for the smallest to the largest diameter class. In contrast to previous records, each entry is contained in a seven-space column. If a particular period is not scheduled for cut, the harvest controls are 0.0 for that period. With a positive random number seed, random values between 0.0 and 1.0 are subsequently assigned to the harvest controls in each of the cutting periods specified on record 4.

In summary, the first five records are one line each. The next two sets of records each have *NGROUP* lines. The last set of records has *NGROUP* • *NUMCYC* lines. Thus, the input file should contain 5 + 2 • *NGROUP* + *NGROUP* • *NUMCYC* lines.

3.2 Prognosis Input Files

In addition to the input file for the optimizer, there are two input files required to run the Prognosis Model: a treelist file and a keyword file (see Wykoff et al. 1982 and Wykoff 1986 for a detailed description of the format for these input files). The treelist file contains a list of tree records that describe attributes of the trees in the initial stand. The keyword file (see fig. 1b, section 4.1, for example) contains all the commands that are required to run Prognosis except for the harvest controls, which are read directly into Prognosis during the optimization. In addition, a response file is also utilized. It simply contains the names of the treelist file and the keyword file, thus telling the Prognosis Model where to find its input data.

3.3 Output File

The output file from the optimization program has four parts (see fig. 1c, section 4.1, for example). Part 1 lists input data including the discount rate, the parameters for the coordinate-search process, the random number seed, the thinning periods, and the species numbers for each species group.

Part 2 lists the present value and step size associated with each coordinate search and pattern search. The first present value represents the evaluation of the initial management regime. When the algorithm terminates, the program writes the present value of the final management regime and the number of calls to Prognosis. The initial value of the stand is calculated, printed, and subtracted from the present value of the regime. This optimal net value is then printed. Note that using net rather than total value in an analysis eliminates problems that might arise if several stands (with different initial volumes) are to be examined. It is clearly necessary to work with net value to avoid double counting if sequential optimizations on the same stand over time are concatenated. (Haight and Monserud [1990b] concatenated three successive 160-year optimizations on the same stand in an attempt to examine the steady-state properties of the optimal any-aged regime.) If volume rather than present value is being optimized, the program also prints average annual production (AAP), which is optimal net volume divided by the length of the regime. Note that AAP equals mean annual increment for an even-aged plantation or stand.

In the third part of the output, the optimal harvest regime is described in three tables (for each species group) that describe various diameter distributions over time: the residual stand, the harvested stand, and the percentage of trees harvested. Four stand averages are written after each diameter distribution: trees per acre, basal area per acre, merchantable volume per acre, and value (dollars per acre). Merchantable volume is either in units of 1,000 board feet or 1,000 cubic feet, whichever was specified on input. Note that the estimate of basal area is biased because it is calculated using the midpoint of the associated diameter class (Prognosis does not use a basal area vector); this estimate is used only for display purposes, however.

In part 4 of the output, the program writes the optimal harvest control variables in exactly the format used for input. This allows for efficiently restarting an optimization from the output of a shorter optimization problem. Part 4 also contains a list of THINDBH keywords specifying the appropriate year, diameter class, species, and harvest proportions for use in the Prognosis model. There is no Prognosis output associated with the optimization. To obtain Prognosis output, these THINDBH keywords must be added to the keyword file for a separate Prognosis simulation. The THINDBH keywords will be written in the optimizer output file if *NOKEY* is not zero (see the seventh entry on record 3 of the optimizer input file described in section 3.1).

4 THREE EXAMPLES

This section gives examples of conversion and plantation management problems that can be solved by the Prognosis Optimization Model. The first two examples assume that we start with a mixed-conifer stand in the *Tsuga heterophylla / Clintonia uniflora* habitat type and seek the best thinning regime for the stand before it is clearcut. Example 1 uses a maximum present value objective and a 20-year horizon. Example 2 uses a maximum cubic foot volume objective and a 40-year horizon. Example 3 assumes that we start with bare ground and plant 600 white pine seedlings per acre and seek the best thinning regime for a 120-year plantation.

4.1 Example 1

Suppose we start with the mixed conifer stand described in the Prognosis User's Guide (Wykoff et al. 1982). The problem is to determine the best thinning in period 1 (year 0) assuming that the stand is clearcut at the beginning of period 3 (year 20). Recall that thinnings take place at the start of a period. The objective is to maximize the present value of the stand over the 20-year horizon using a 4 percent discount rate. Trees less than 6 inches are not merchantable and cost $0.10 per tree to cut. Trees greater than 6 inches are worth $100.00 per Mbf independent of species and diameter. Harvest controls are defined for diameter classes, and all species are included in one species group.

The optimizer input file for this problem is listed in figure 1a; here we highlight some of the entries. Because the random number seed (record 1, entry 2) is positive, random proportions will be given to the starting values for the harvest controls. Because *NUMCYC = 2* (record 3, entry 1), there are two growth periods in the planning horizon. Because *MERCH = 4* (record 3, entry 2), there are three unmerchantable diameter classes. There is one species group (record 3, entry 3), and merchantable board foot volume is used to value harvested trees (record 3, entry 4). The length of the growth period is 10 years (record 3, entry 5). There is one cutting period (record 4, entry 1), which is the first period (record 4, entry 2). Record 5 gives the diameter class bounds; note that there are eight diameter classes. Record 6 indicates one species in the group, and its species code is the shortcut 0 (all species are lumped into one group). Record 7 gives the tree prices (unmerchantable diameter classes) and stumpage prices (merchantable diameter classes). Records 8 and 9 list the initial values for the harvest controls for periods 1 and 2. Note that period 2 harvest controls equal zero because no harvest is scheduled. Although harvest controls for period 1 equal 0, they are subsequently assigned random proportions between 0.0 and 1.0 because harvesting was specified for the first period.

The keyword file used to address Prognosis during the optimization run is listed in figure 1b. The first four lines are based on the stand information given in the Prognosis User's Guide. The NUMTRIP command specifies record tripling in the first two periods.[1] There are two growth projection periods (NUMCYCLE is 2). The regeneration establishment routine is called in the first period (ESTAB is 0). Within this routine, the number of plot replications is 10, and no output is requested. The TREEDATA keyword indicates that Prognosis should look for a treelist file for the description of the initial stand. The last two keywords are required to end the file.

This optimization run takes about 160 seconds on an IBM PS/2 Model 80 with a 20-Mhz processor. During the optimization, there are 31 calls to the Prognosis Model for projection. Output from this example is listed in figure 1c. The optimal strategy for maximizing present value is simple: remove all merchantable trees immediately. The regime produces $216.53 after subtracting the initial value of the stand. Because the optimizer seeks diameter-class harvest controls in only one species group and period, the solution is obtained with a relatively short execution time. Beginning this problem with random harvest controls generated by any integer seed between 1 and 6 results in the same solution.

[1]Prognosis is well-behaved if it has approximately 500-600 tree records to project (W. Wykoff, personal communication). Normal use of the NUMTRIP keyword (tripling twice) is intended to generate adequate variation from stand inventories that contain fewer tree records. We strongly recommend against using the optimizer with the NUMTRIP keyword disabled. If few records are projected, the optimizer may devise an anomalous management regime that favors the few fast growing tree records—which will unrealistically represent a large number of trees per acre. We also recommend using 5-year projection periods until version 6 of the Prognosis Model is released. This modification is necessary to eliminate a height-growth bias on small trees.

Figure 1a—Optimizer input file *FIG1.INP* for example 1.

```
   1   1.   2.
 .04   .1  1.0  0.2 999.
   2   4    1    1   10    1    1
   1   1
   9   0.   2.   4.   7.  10.  14.  18.  22.  40.
   1   0.
 -.1  -.1  -.1  100. 100. 100. 100. 100.
0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
```

Figure 1b—Prognosis keyword file *FIG1.KEY* for example 1.

```
STDIDENT
S248112   HOOKE AND JEEVES DISCRETE STEP; FIGURE 1
DESIGN                                           11.0       1.0
STDINFO         18.0      570.0       0.00        8.0       3.0        34.0
NUMTRIP           2.
NUMCYCLE          2.
ESTAB             0.
MINREP           10.
OUTPUT            0.
END
TREEDATA
PROCESS
STOP
```

Figure 1c—Optimizer output file *FIG1.OUT* for example 1.

```
BOBS2: PROGNOSIS OPTIMIZATION BY THE BOBS

OPTIMIZATION NUMBER        1
INTEREST RATE =            0.040
ACCELERATION STEP =        0.100
BEGINNING STEP SIZE =      1.000
MINIMUM STEP SIZE =        0.200
MINIMUM GAIN =           999.000
RANDOM NUMBER SEED =       1.000

THE OPTIMIZER WILL DETERMINE 1 HARVEST, OCCURRING IN PERIOD  1

SPECIES CODES FOR GROUP 1 ARE  0.

HOOKE & JEEVES COORDINATE SEARCH PROCESS:

PRESENT VALUE =     386.84
 PERFORMED STEP SEARCH WITH DELTA =  1.000
PRESENT VALUE =     588.74
  ACCELERATION STEP NOT SUCCESSFUL
 PERFORMED STEP SEARCH WITH DELTA =  0.500
PRESENT VALUE =     588.74
  ACCELERATION STEP NOT SUCCESSFUL
 PERFORMED STEP SEARCH WITH DELTA =  0.250
PRESENT VALUE =     588.74
  ACCELERATION STEP NOT SUCCESSFUL
 PERFORMED STEP SEARCH WITH DELTA =  0.125
PRESENT VALUE =     588.74

NUMBER OF PROGNOSIS SIMULATIONS =   31

INITIAL VALUE =      372.21 DOLLARS/AC

OPTIMAL PRESENT NET VALUE (PNV) =     216.53 DOLLARS/AC


RESIDUAL TREES PER ACRE FOR SPECIES GROUP  1

 DBH      --------YEAR------->
CLASS     0      10      20
  1.   245.5  929.8     0.0
  3.    54.5   10.9     0.0
  5.   136.6  132.6     0.0
  8.     0.0   85.1     0.0
 12.     0.0    8.7     0.0
 16.     0.0    0.0     0.0
 20.     0.0    0.0     0.0
 31.     0.0    0.0     0.0
TOTAL   437.  1167.      0.
BA/AC    27.    68.      0.
VO/AC   0.00   1.87    0.00
$$/AC  -43.7   79.8     0.0

HARVESTED TREES PER ACRE FOR SPECIES GROUP  1:

 DBH      --------YEAR------->
CLASS     0      10      20
  1.     0.0    0.0  2006.8
  3.     0.0    0.0     7.4
  5.     0.0    0.0    56.2
  8.    78.5    0.0   113.3
 12.    21.0    0.0    60.7
 16.     0.0    0.0     2.2
 20.     0.0    0.0     0.0
 31.     0.0    0.0     0.0
TOTAL    99.     0.   2246.
BA/AC    47.     0.    116.
VO/AC   4.16   0.00    5.86
$$/AC  415.9    0.0   378.8

PERCENTAGE TREES PER ACRE CUT FOR SPECIES GROUP  1:

 DBH      --------YEAR------->
CLASS     0       10      20
  1.     0.00    0.00  100.00
  3.     0.00    0.00  100.00
  5.     0.00    0.00  100.00
  8.   100.00    0.00  100.00
 12.   100.00    0.00  100.00
 16.     0.00    0.00  100.00
 20.     0.00    0.00    0.00
 31.     0.00    0.00    0.00

OPTIMAL HARVEST CONTROL PARAMETERS FOR SPECIES GROUP  1:
 0.0000 0.0000 0.0000 1.0000 1.0000 0.0000 0.0000 0.0000
 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000

OPTIMAL HARVEST KEYWORDS FOR PROGNOSIS
 (INSERT IN THE KEYWORD FILE)
THINDBH            1.        7.        10.     1.0000
THINDBH            1.       10.        14.     1.0000
```

4.2 Example 2

In this case we start with the same stand, but the problem is to determine the best thinnings in period 1 (year 0) and 3 (year 20) assuming that the stand is clearcut at the beginning of period 5 (year 40). The initial stand is approximately 50 years old. The objective is to maximize the merchantable cubic foot volume produced in the stand over the 40-year horizon. Trees less than 6 inches in diameter are not merchantable. Harvest controls are defined for diameter classes in two species groups. Group 1 includes the two intolerant species, western larch (*Larix occidentalis* Nutt.) and lodgepole pine (*Pinus contorta* Dougl. ex Loud.), and group 2 includes all other species. (Note that we use two species groups for illustration. Because there are no stumpage price differences by species group, superior solutions may be obtained with only one group.)

The input file for the optimizer is listed in figure 2a; note the differences from the input listed in figure 1a. Because we are maximizing merchantable cubic foot volume, the discount rate is 0.0 (record 2, entry 1). With a 10-year projection interval and a 40-year horizon, *NUMCYC = 4* (record 3, entry 1). There are two species groups (record 3, entry 3), and the volume code (record 3, entry 4) is less than zero to indicate a cubic foot volume measure. Record 4 indicates that there are two harvests, in periods 1 and 3. Records 6 and 7 give the number of species and species code for each of the two species groups. Note that there are two lines of prices (records 8 and 9), one for each species group. Unmerchantable tree classes have no value; merchantable tree prices equal 1, so that value is measured in thousand cubic foot units. Finally, note that initial harvest control variable values are entered as a block for each species group. The zeros in periods 1 and 3 will be replaced with random harvest proportions because the random seed is positive.

The Prognosis keyword file for the optimization run is listed in figure 2b. In this case there are four growth projection periods (NUMCYCLE is 4). The regeneration establishment routine is called at the start of periods 1 (year 0) and 3 (year 20).

The optimization run requires about 30 minutes on an IBM PS/2 Model 80 with a 20-Mhz processor. During the optimization, there are 119 calls to the Prognosis Model for stand projection. The output is listed in figure 2c. The optimal regime calls for removing almost all of the merchantable intolerants immediately. The second harvest (year 20) includes a precommercial thinning and the removal of a few of the largest trees in both species groups. In the clearcut in year 40, just over 200 merchantable trees per acre are harvested. After subtracting the initial volume, the stand produces 116 ft^3 per acre per year. As we are about to see in the next example, the stand is obviously understocked.

Figure 2a—Optimizer input file *FIG2.INP* for example 2.

```
 1   5.    6.    7.    8.    9.   10.
 .0   .1  1.0   .2  999.
 4    4     2   -1    10     1     1
 2    1     3
 9   0.    2.    4.    7.   10.   14.   18.   22.   40.
 2   2.    7.
 5   1.    3.    4.    5.    6.
 0.   0.    0.    1.    1.    1.    1.    1.
 0.   0.    0.    1.    1.    1.    1.    1.
0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
```

Figure 2b—Prognosis keyword file *FIG2.KEY* for example 2.

```
STDIDENT
S248112    HOOKE AND JEEVES DISCRETE STEP: FIGURE 2
DESIGN                                           11.0        1.0
STDINFO          18.0       570.0        0.00     8.0        3.0       34.0
NUMTRIP           2.
NUMCYCLE          4.
ESTAB             0.
MINREP           10.
OUTPUT            0.
END
ESTAB            20.
END
TREEDATA
PROCESS
STOP
```

Figure 2c—Optimizer output file *FIG2.OUT* for example 2.

```
BOBS2: PROGNOSIS OPTIMIZATION BY THE BOBS

OPTIMIZATION NUMBER      1
INTEREST RATE =          0.000
ACCELERATION STEP =      0.100
BEGINNING STEP SIZE =    1.000
MINIMUM STEP SIZE =      0.200
MINIMUM GAIN =         999.000
RANDOM NUMBER SEED =     5.000

THE OPTIMIZER WILL DETERMINE  2 HARVESTS.
THEY OCCUR IN PERIODS:   1   3

SPECIES CODES FOR GROUP 1 ARE  2.  7.
SPECIES CODES FOR GROUP 2 ARE  1.  3.  4.  5.  6.

HOOKE & JEEVES COORDINATE SEARCH PROCESS:

PRESENT VALUE =      3.57
 PERFORMED STEP SEARCH WITH DELTA =  1.000
PRESENT VALUE =      5.57
  ACCELERATION STEP NOT SUCCESSFUL
 PERFORMED STEP SEARCH WITH DELTA =  0.500
PRESENT VALUE =      5.57
  ACCELERATION STEP SUCCESSFUL
PRESENT VALUE =      5.57
 PERFORMED STEP SEARCH WITH DELTA =  0.250
PRESENT VALUE =      5.57
  ACCELERATION STEP NOT SUCCESSFUL
 PERFORMED STEP SEARCH WITH DELTA =  0.125
PRESENT VALUE =      5.66

NUMBER OF PROGNOSIS SIMULATIONS =  119

INITIAL VOLUME =              1019.8 CUFT/AC

NET OPTIMAL VOLUME =          4635.5 CUFT/AC

AVERAGE ANNUAL PRODUCTION =  115.9 CUFT/AC/YR


RESIDUAL TREES PER ACRE FOR SPECIES GROUP  1

 DBH     --------YEAR------->
CLASS    0     10     20     30     40
  1.    0.0   30.5    1.9    0.9    0.0
  3.    0.0    0.0    0.0    0.0    0.0
  5.   15.8    0.0    0.0    0.0    0.0
  8.    0.0   15.4    8.2    1.5    0.0
 12.    4.4    4.1    7.8    8.9    0.0
 16.    0.0    0.0    0.0    4.8    0.0
 20.    0.0    0.0    0.0    0.0    0.0
 31.    0.0    0.0    0.0    0.0    0.0
TOTAL   20.    50.    18.    16.     0.
BA/AC    6.     9.     9.    14.     0.
VO/AC  0.11   0.21   0.22   0.34   0.00
$$/AC   0.1    0.2    0.2    0.3    0.0

RESIDUAL TREES PER ACRE FOR SPECIES GROUP  2
 DBH     --------YEAR------->
CLASS    0      10      20      30     40
  1.   245.5  863.9 1800.9 1415.1    0.0
  3.    54.5   42.0    5.4   39.6    0.0
  5.   120.8  107.0   79.0   33.5    0.0
  8.    11.2   71.3   98.5   81.4    0.0
 12.    15.9   21.2   40.7   74.4    0.0
 16.     0.0    4.9    7.1   28.9    0.0
 20.     0.0    0.0    0.0    1.8    0.0
 31.     0.0    0.0    0.0    0.0    0.0
TOTAL   448.  1110.  2032.  1675.     0.
BA/AC    41.    76.   104.   150.     0.
VO/AC   0.39   0.99   1.61   2.82   0.00
$$/AC    0.4    1.0    1.6    2.8    0.0

HARVESTED TREES PER ACRE FOR SPECIES GROUP  1:

 DBH      --------YEAR------->
CLASS     0     10     20     30     40
  1.     0.0    0.0   33.2    0.0   13.7
  3.     0.0    0.0    0.0    0.0    0.0
  5.     0.0    0.0    0.0    0.0    0.0
  8.    64.3    0.0    0.0    0.0    0.0
 12.     0.6    0.0    1.1    0.0    3.5
 16.     0.0    0.0    1.4    0.0   10.7
 20.     0.0    0.0    0.0    0.0    0.0
 31.     0.0    0.0    0.0    0.0    0.0
TOTAL    65.     0.    36.     0.    28.
BA/AC    26.     0.     3.     0.    18.
VO/AC   0.49   0.00   0.09   0.00   0.43
$$/AC    0.5    0.0    0.1    0.0    0.4

HARVESTED TREES PER ACRE FOR SPECIES GROUP  2:

 DBH      --------YEAR------->
CLASS     0     10     20     30     40
  1.     0.0    0.0  213.2    0.0 1568.0
  3.     0.0    0.0    0.8    0.0   84.5
  5.     0.0    0.0    0.0    0.0    8.7
  8.     3.0    0.0    0.0    0.0   73.8
 12.     0.0    0.0    0.0    0.0   71.5
 16.     0.0    0.0    7.3    0.0   41.7
 20.     0.0    0.0    0.6    0.0    9.9
 31.     0.0    0.0    0.0    0.0    0.4
TOTAL     3.     0.   222.     0.  1859.
BA/AC     1.     0.    13.     0.   181.
VO/AC   0.03   0.00   0.32   0.00   4.30
$$/AC    0.0    0.0    0.3    0.0    4.3

PERCENTAGE TREES PER ACRE CUT FOR SPECIES GROUP  1:

 DBH      --------YEAR------->
CLASS     0      10     20      30     40
  1.     0.00   0.00  94.65   0.00 100.00
  3.     0.00   0.00   0.00   0.00 100.00
  5.     0.00   0.00   0.00   0.00   0.00
  8.   100.00   0.00   0.00   0.00   0.00
 12.    12.50   0.00  12.50   0.00 100.00
 16.     0.00   0.00 100.00   0.00 100.00
 20.     0.00   0.00   0.00   0.00   0.00
 31.     0.00   0.00   0.00   0.00   0.00
```

Figure 2c (Con.)

PERCENTAGE TREES PER ACRE CUT FOR SPECIES GROUP 2:

DBH CLASS	YEAR 0	10	20	30	40
1.	0.00	0.00	10.58	0.00	100.00
3.	0.00	0.00	12.50	0.00	100.00
5.	0.00	0.00	0.00	0.00	100.00
8.	20.92	0.00	0.00	0.00	100.00
12.	0.00	0.00	0.00	0.00	100.00
16.	0.00	0.00	50.60	0.00	100.00
20.	0.00	0.00	100.00	0.00	100.00
31.	0.00	0.00	0.00	0.00	100.00

OPTIMAL HARVEST CONTROL PARAMETERS FOR SPECIES GROUP 1:

0.0000	0.0000	0.0000	1.0000	0.1250	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.9465	0.0000	0.0000	0.0000	0.1250	1.0000	0.0000	0.0000
0.0000	D.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

OPTIMAL HARVEST CONTROL PARAMETERS FOR SPECIES GROUP 2:

0.0000	0.0000	0.0000	0.2092	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.1058	0.1250	0.000D	0.0000	0.0000	0.5060	1.0000	0.0000
0.0000	0.00DD	0.0DDD	0.00D0	0.0000	0.0000	0.0000	0.0000

OPTIMAL HARVEST KEYWORDS FOR PROGNOSIS
(INSERT IN THE KEYWORD FILE)

THINDBH	1.	7.	10.	1.0000	2.
THINDBH	1.	7.	10.	1.0000	7.
THINDBH	1.	7.	10.	0.2092	1.
THINDBH	1.	7.	10.	0.2092	3.
THINDBH	1.	7.	10.	0.2092	4.
THINDBH	1.	7.	10.	0.2092	5.
THINDBH	1.	7.	10.	0.2092	6.
THINDBH	1.	10.	14.	0.1250	2.
THINDBH	1.	10.	14.	0.1250	7.
THINDBH	21.	0.	2.	0.9465	2.
THINDBH	21.	0.	2.	0.9465	7.
THINDBH	21.	0.	2.	0.1058	1.
THINDBH	21.	0.	2.	0.1058	3.
THINDBH	21.	0.	2.	0.1058	4.
THINDBH	21.	0.	2.	0.1058	5.
THINDBH	21.	0.	2.	0.1058	6.
THINDBH	21.	2.	4.	0.1250	1.
THINDBH	21.	2.	4.	0.1250	3.
THINDBH	21.	2.	4.	0.1250	4.
THINDBH	21.	2.	4.	0.1250	5.
THINDBH	21.	2.	4.	0.1250	6.
THINDBH	21.	10.	14.	0.1250	2.
THINDBH	21.	10.	14.	0.1250	7.
THINDBH	21.	14.	18.	1.0000	2.
THINDBH	21.	14.	18.	1.0000	7.
THINDBH	21.	14.	18.	0.5060	1.
THINDBH	21.	14.	18.	0.5060	3.
THINDBH	21.	14.	18.	0.5060	4.
THINDBH	21.	14.	18.	0.5060	5.
THINDBH	21.	14.	18.	0.5060	6.
THINDBH	21.	18.	22.	1.0000	1.
THINDBH	21.	18.	22.	1.0000	3.
THINDBH	21.	18.	22.	1.0000	4.
THINDBH	21.	18.	22.	1.0000	5.
THINDBH	21.	18.	22.	1.0000	6.

4.3 Example 3

The following example is used by Haight and Monserud (1990b) to develop optimal white pine plantation management regimes. The objective is to maximize cubic foot volume production over a 120-year horizon in which thinnings may take place every 20 years beginning in year 20. The fixed silvicultural inputs are defined in the Prognosis keyword file (fig. 3b). Using the same stand and site parameters as in the previous example (see the STDINFO keyword), this run begins with bare ground and plants 600 white pine seedlings per acre after site preparation is completed. During the 120-year projection (24 cycles of 5 years), the regeneration establishment model is called if thinning removes at least 20 ft^2 per acre. If the thinning leaves less than 80 ft^2 per acre then a regeneration cut is assumed to have occurred; site preparation is performed, and the establishment and development of both advanced and subsequent regeneration are simulated. If the thinning leaves at least 80 ft^2 per acre then site preparation is not done, and only advanced regeneration is simulated. Because we start with bare ground (note the NOTREES keyword), we have taken out the TREEDATA keyword.

The optimizer input file is listed in figure 3a. Unlike the previous examples, this optimization does not start with random harvest controls (on record 1, entry 2 is negative). Instead, the run begins with the optimal 100-year regime specified in the block of harvest controls at the end of the file. When the optimization problem is complicated, this sequential strategy for specifying initial conditions is far more efficient than beginning with many random starts.

This optimization problem requires 177 calls to Prognosis and takes 112 minutes on a 20-Mhz personal computer. Output describing the optimal plantation management is listed in figure 3c. The optimal regime, which produces 163 ft^3 per acre per year, is discussed in detail by Haight and Monserud (1990b).

Figure 3a—Optimizer input file *FIG3.INP* for example 3.

```
   1  -1.
  .0   .1  1.0  .2  999.
  24    4    1   -1    5    2    1
   5    5    9   13   17   21
   9   0.   2.   4.   7.  10.  14.  18.  22.  40.
   1   0.
  0.   0.   0.   1.   1.   1.   1.   1.
0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
1.0000 0.5262 0.0224 0.8346 0.0000 0.0000 0.0000 0.0000
0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
0.1250 0.0000 0.0000 0.0000 0.1250 0.5000 0.0000 0.0000
0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
1.0000 0.0000 0.0000 0.7500 0.0000 0.0000 0.3750 0.0000
0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
0.0000 0.0000 0.0000 0.8750 1.0000 0.0000 0.0000 0.0000
0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
```

Figure 3b—Prognosis keyword file *FIG3.KEY* for example 3.

```
STDIDENT
S248112        HOOKE AND JEEVES;   PLANT 600 WWP & GROW 120 YEARS
STDINFO          18.0      570.0      0.00       8.0       3.0       34.0
NUMTRIP           5.
NOTREES
TIMEINT                     5.
NUMCYCLE         24.
ESTAB             0.
PLANT             2.        1.        600.
MECHPREP          1.       50.
BURNPREP          1.       30.
MINREP           10.
OUTPUT            0.
END
COMMENT
Schedule a normal call to ESTAB whenever a regeneration cut occurs:
END
IF
 (FRAC((CYCLE-1)/4 ) EQ 0) AND (CYCLE NE 1)        &
  AND ( ((BBA-ABA) GT 20) AND (ABA LT 80) )
THEN
ESTAB
MECHPREP          0.       20.
END
ENDIF
COMMENT
Otherwise pump in advanced regeneration every 20 years:
END
IF
 (FRAC((CYCLE-1)/4 ) EQ 0) AND (CYCLE NE 1)         &
  AND  ( ((BBA-ABA) LE 20) OR (ABA GE 80) )
THEN
ESTAB
MECHPREP          0.        0.
TALLYONE          3.
END
ENDIF
PROCESS
STOP
```

Figure 3c—Optimizer output file *FIG3.OUT* for example 3.

BOBS2: PROGNOSIS OPTIMIZATION BY THE BOBS

OPTIMIZATION NUMBER 1
INTEREST RATE = 0.000
ACCELERATION STEP = 0.100
BEGINNING STEP SIZE = 1.000
MINIMUM STEP SIZE = 0.200
MINIMUM GAIN = 999.000
RANDOM NUMBER SEED = -1.000

THE OPTIMIZER WILL DETERMINE 5 HARVESTS.
THEY OCCUR IN PERIODS: 5 9 13 17 21

SPECIES CODES FOR GROUP 1 ARE 0.

HOOKE & JEEVES COORDINATE SEARCH PROCESS:

PRESENT VALUE = 19.10
PERFORMED STEP SEARCH WITH DELTA = 1.000
PRESENT VALUE = 19.47
ACCELERATION STEP NOT SUCCESSFUL
PERFORMED STEP SEARCH WITH DELTA = 0.500
PRESENT VALUE = 19.53
ACCELERATION STEP SUCCESSFUL
PRESENT VALUE = 19.53
PERFORMED STEP SEARCH WITH DELTA = 0.250
PRESENT VALUE = 19.53
ACCELERATION STEP NOT SUCCESSFUL
PERFORMED STEP SEARCH WITH DELTA = 0.125
PRESENT VALUE = 19.53

NUMBER OF PROGNOSIS SIMULATIONS = 177

INITIAL VOLUME = 0.0 CUFT/AC

NET OPTIMAL VOLUME = 19531.1 CUFT/AC

AVERAGE ANNUAL PRODUCTION = 162.8 CUFT/AC/YR

RESIDUAL TREES PER ACRE FOR SPECIES GROUP 1

DBH	--------YEAR------->											
CLASS	0	10	20	30	40	50	60	70	80	90	100	110	120
1.	0.0	1976.8	0.0	1950.2	2871.6	2441.1	0.0	686.6	609.2	866.3	725.0	730.5	0.0
3.	0.0	0.0	156.1	22.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	11.9	0.0
5.	0.0	0.0	249.9	197.9	92.3	33.8	2.1	0.6	0.0	0.0	0.0	0.0	0.0
8.	0.0	0.0	1.1	155.8	198.3	135.5	21.9	14.5	6.0	2.3	0.0	0.0	0.0
12.	0.0	0.0	0.0	11.9	69.3	143.1	155.9	105.3	0.0	2.9	1.8	1.4	0.0
16.	0.0	0.0	0.0	0.0	2.6	22.2	53.6	65.8	65.0	45.1	23.7	12.0	0.0
20.	0.0	0.0	0.0	0.0	0.0	1.0	2.7	20.9	40.7	41.8	46.5	36.4	0.0
31.	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.8	1.4	14.3	25.7	41.7	0.0
TOTAL	0.	1977.	407.	2338.	3234.	2777.	236.	894.	722.	973.	823.	834.	0.
BA/AC	0.	11.	49.	115.	167.	218.	212.	234.	193.	237.	275.	321.	0.
VO/AC	0.00	0.00	0.00	1.18	3.21	5.44	7.02	9.03	8.33	10.49	12.35	14.34	0.00
$$/AC	0.0	0.0	0.0	1.2	3.2	5.4	7.0	9.0	8.3	10.5	12.4	14.3	0.0

HARVESTED TREES PER ACRE FOR SPECIES GROUP 1:

DBH	--------YEAR------->											
CLASS	0	10	20	30	40	50	60	70	80	90	100	110	120
1.	0.0	0.0	3221.8	0.0	0.0	0.0	1986.5	0.0	0.0	0.0	0.0	0.0	485.6

Figure 3c (Con.)

3.	0.0	0.0	173.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	111.2
5.	0.0	0.0	5.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8.	0.0	0.0	5.7	0.0	0.0	0.0	65.8	0.0	0.9	0.0	0.6	0.0	0.0
12.	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	74.3	0.0	2.2	0.0	1.1
16.	0.0	0.0	0.0	0.0	2.6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.1
20.	0.0	0.0	0.0	0.0	0.0	0.0	1.6	0.0	0.0	0.0	0.0	0.0	29.9
31.	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	52.0
TOTAL	0.	0.	3407.	0.	3.	0.	2054.	0.	75.	0.	3.	0.	683.
BA/AC	0.	0.	29.	0.	4.	0.	40.	0.	59.	0.	2.	0.	351.
VO/AC	0.00	0.00	0.02	0.00	0.10	0.00	0.74	0.00	2.62	0.00	0.07	0.00	15.99
$$/AC	0.0	0.0	0.0	0.0	0.1	0.0	0.7	0.0	2.6	0.0	0.1	0.0	16.0

PERCENTAGE TREES PER ACRE CUT FOR SPECIES GROUP 1:

DBH CLASS	--------YEAR-------> 0	10	20	30	40	50	60	70	80	90	100	110	120
1.	0.00	0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00	100.00
3.	0.00	0.00	52.62	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00
5.	0.00	0.00	2.24	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
8.	0.00	0.00	83.46	0.00	0.00	0.00	75.00	0.00	12.50	0.00	100.00	0.00	0.00
12.	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00	0.00	55.00	0.00	100.00
16.	0.00	0.00	0.00	0.00	50.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00
20.	0.00	0.00	0.00	0.00	0.00	0.00	37.50	0.00	0.00	0.00	0.00	0.00	100.00
31.	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00

OPTIMAL HARVEST CONTROL PARAMETERS FOR SPECIES GROUP 1:

```
0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
1.0000 0.5262 0.0224 0.8346 0.0000 0.0000 0.0000 0.0000
0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
0.0000 0.0000 0.0000 0.0000 0.0000 0.5000 0.0000 0.0000
0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
1.0000 0.0000 0.0000 0.7500 0.0000 0.0000 0.3750 0.0000
0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
0.0000 0.0000 0.0000 0.1250 1.0000 0.0000 0.0000 0.0000
0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
0.0000 0.0000 0.0000 1.0000 0.5500 0.0000 0.0000 0.0000
0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
```

OPTIMAL HARVEST KEYWORDS FOR PROGNOSIS
(INSERT IN THE KEYWORD FILE)

```
THINDBH        21.      0.      2.    1.0000
THINDBH        21.      2.      4.    0.5262
THINDBH        21.      4.      7.    0.0224
THINDBH        21.      7.     10.    0.8346
THINDBH        41.     14.     18.    0.5000
THINDBH        61.      0.      2.    1.0000
THINDBH        61.      7.     10.    0.7500
THINDBH        61.     18.     22.    0.3750
THINDBH        81.      7.     10.    0.1250
THINDBH        81.     10.     14.    1.0000
THINDBH       101.      7.     10.    1.0000
THINDBH       101.     10.     14.    0.5500
```

5 PROGRAM EXECUTION AND SIZE LIMITS

As we described in sections 3.1 and 3.2, there are several files required to run the Prognosis Optimization Model. Here we describe how to organize these files and execute the optimizer. File names correspond to those contained in the distribution diskette. Installing the program is the subject of section 6.

We recommend executing the Prognosis Optimization Model with a DOS batch file such as *FIG1.BAT* with the following commands:

```
DT
COPY FIG1.INP   BOBS.INP
COPY FIG1.KEY   TEST.KEY
OPTGROW < TEST.RSP
COPY BOBS.OUT   FIG1.OUT
DT
```

FIG1.BAT contains the set of commands needed to solve the harvest problem discussed in example 1 above. The DT execute commands print the date and time at the start and end of the optimization. File *FIG1.INP* contains the input data for the optimizer (fig. 1a). It is copied to file *BOBS.INP*, which the optimizer will read for input parameters. File *FIG1.KEY* contains the Prognosis keywords (fig. 1b), and it is copied to file *TEST.KEY*, which Prognosis will read for keywords. The command *OPTGROW* executes the optimizer. This command points to response file *TEST.RSP*, which contains file names for the Prognosis treelist file *TEST.TRE* and keyword file *TEST.KEY*. After executing the *OPTGROW* command, the output file *BOBS.OUT* is copied to *FIG1.OUT* (fig. 1c), which will not be written over by a subsequent execution of the optimizer (as *BOBS.OUT* will be).

Now that we know what is contained in the input files, it is time to make the optimization run. To do this, simply type the command:

FIG1

which initiates execution of the batch file that contains the commands for running the program.

Limits to the size of the optimization problem that can be solved are affected by the size of the arrays declared in the optimization program (see the appendix; Haight and Monserud [1990a] also discuss practical limits on the complexity of optimization problems that can be solved). Currently the maximum numbers of periods, diameter classes, and species groups are 33, 8, and 3, respectively. If these are changed, be sure to modify array size declarations in each of the following optimization routines: MAIN and subroutines CONTROL, OUT, STATE, VALUE, and GKEY (all of which are in *OPT.FOR*), and PMAIN2 (which is in *PMAIN.FOR*).

Limits to the size of the problem are also affected by the maximum number of keywords (1,200) and keyword parameters (6,000) that may be processed by the Prognosis Model. In a keyword file for a thinning regime, THINDBH keywords take up the most space. Whenever harvesting takes place by species group, a THINDBH keyword is created for each diameter class, tree species, and period in which harvesting may occur. Further, there are five parameters on each THINDBH keyword. To see how rapidly these accumulate, suppose we define an optimization problem with 10 tree species among three groups, eight diameter classes, and 15 cutting periods. Then, there are 10 x 8 x 15 = 1,200 keywords and 1,200 x 5 = 6,000 parameters. Because there are several other keywords that define the harvest regime (ESTAB, for example), this keyword file is too large for the Prognosis Model, with uncertain consequences. The Prognosis Model may either terminate prematurely or not process keywords beyond the upper limit. Thus, it is important to make sure the problem size is within these boundaries.

Execution time depends on the size of the problem. Example 1, which includes one thinning period, eight diameter classes, one species group, and a 20-year horizon, requires about 3 minutes on a 386 personal computer (an IBM PS/2 Model 80) with a 20-Mhz processor. Problems with thinnings every 20 years and 40-, 60-, and 160-year time horizons require around 25, 50, and 240 minutes, respectively. For more than one species group, multiply these times by the number of groups. Execution times on other personal computers should be approximately proportional to the ratio of the processor frequencies. For example, the 3-minute run discussed above took 3.8 minutes on a Compaq 386 with a 16-Mhz clock.

6 PROGRAM INSTALLATION

There are two ways to get an executable version of the optimizer on your system. The first—and by far the simplest—is to extract it from the archived executable code stored on the distribution diskette. This approach is fine if you simply want to run the optimizer and do not need to modify anything in the program. This section explains this choice. The second way to get an executable program is to build it from the source code contained on the distribution diskette. This procedure is explained in section 8.

For openers, we assume that MS-DOS is the operating system on your personal computer (we will occasionally use the DOS wildcard character *). Begin by copying *OPTGROW.ARC*, *ARC.DOC*, and *ARC.EXE* files from the program diskette onto a directory dedicated to running the optimizer. The Prognosis Optimization Model and input files for three examples are compactly written onto *OPTGROW.ARC* using a public domain archiving utility called ARC that is distributed by Systems Enhancement Associates, Wayne, NJ. This utility is stored in file *ARC.EXE* with documentation given in *ARC.DOC*. The *SOURCE*.ARC* files on the distribution diskette contain the source code for the Prognosis optimizer and are needed only if you plan to create a new executable program. Unload all files from *OPTGROW.ARC* with the ARC extraction command:

ARC X OPTGROW

The extracted executable module, *OPTGROW.EXE*, will require about 529K. The three overlays are *ESESTAB.OVL*, *ESGROWTH.OVL*, and *ESINPUT.OVL*. You will find that the following input and batch files also have been extracted: *TEST.RSP*, *TEST.TRE*, *FIG*.KEY*, *FIG*.INP*, *FIG*.BAT*, *DT.EXE*, and *UNLOAD.BAT*.

These are all the files needed to run the example problems. (Note that the last file is used only if you are going to extract the source code as discussed in section 8.)

7 PROGRAM STRUCTURE

This section describes the structure of the source code for the Prognosis Optimization Model, which is listed in the appendix. The driver program for stand optimization (program MAIN) contains a coordinate-search process that repeatedly evaluates and attempts to improve the value of the harvest regime for a given stand. The harvest regime specifies the proportions (fractions between 0.0 and 1.0) of trees cut by size class, species group, and period. These proportions are called "harvest controls." During the course of an optimization the Prognosis Model is called as a subroutine to project the stand as a function of the harvest controls. Except where noted, no changes have been made in the Prognosis code. The variables in MAIN are independent of those in Prognosis and are defined in the program listing. The MAIN program contains three blocks: initialization, search process, and writing the results. The first two blocks are described here, and the output block is described in section 3.3. When reading the descriptions of these blocks, make sure to look at the source code and comments listed in the appendix.

7.1 Initialization

The initialization phase involves reading input parameters that control the optimization, opening the Prognosis files, initializing the harvest controls, and evaluating the harvest regime.

The optimization begins by reading control parameters for the optimizer (see fig. 1a) on unit 25. These parameters are discussed in detail in section 3.1. Among other things, they include the diameter classes, species groups, and periods in which harvesting may take place, the stumpage prices used to value harvests, and the initial values given to the harvest controls. After optimizer initialization, an outer loop is entered that repeats the optimization run with different seeds for generating random starting points for the harvest controls.

The next step involves opening the Prognosis files. Before we describe this step, a little background information is needed. During the course of the optimization, the Prognosis Model is repeatedly called as a subroutine to project candidate management regimes. These simulations are controlled by the keyword file (unit 5) and the treelist file (unit 2), which must be present in addition to the input file for the optimizer (see section 5). The keyword file contains all the controls for the Prognosis Model except for the harvest controls. These are not specified as keywords but instead are copied directly into Prognosis by using the OPADD routine described below. The treelist file contains the initial stand inventory. Because MAIN contains a loop that allows more than one optimization to be done in a given run, units 2 and 5 must be rewound for each optimization.

The driver program for the Prognosis Model is split into three subprograms: PMAIN1, PMAIN2, and PMAIN3. The first subprogram contains the Prognosis initialization routines, including file-open statements, and it is called first. The file-open routines are near the top of PMAIN1, and they are called only once (when *ISKIP=0*) during the program execution. The file-open routines are skipped for each subsequent call to PMAIN1 (*ISKIP=1*).

After calling PMAIN1 and when test variable *JSET*<0, all the Prognosis variables necessary to restart the simulation from that point are stored on a scratch file (unit 50) by calling BHPUT. This is done so that subsequent Prognosis calls during the coordinate search do not have to repeat PMAIN1. When *JSET*>0, *JSET* is the number of periods before the first harvest. In this case, the stand is grown *JSET* periods by calling PMAIN2. The resulting Prognosis variables are stored on the scratch file by calling BHPUT in PMAIN2. Again, this avoids repeating the initial projection before harvests can take place during the coordinate search. Think of the optimization as a search of a multibranched decision tree. The optimizer uses BHPUT (and BHGET, which reverses the process) to back down the decision tree only as far as necessary to reach another node or branch in its search for the highest point. (Our implementation of BHPUT and BHGET is high-graded from the Parallel Processing Extension of Crookston and Stage [1988].) Note that the optimizer would execute faster if we used BHPUT to store the status of Prognosis in dynamic memory rather than in a scratch file. Unfortunately, there is not enough memory available within the 640K limitation of MS-DOS to do this.

The third section of the original Prognosis driver, PMAIN3, contains calls to output routines. These are not called by the optimization program because they are not needed and would seriously slow down execution of the optimizer. If Prognosis output is desired for a particular optimal policy, the user simply runs the normal Prognosis program (rather than the optimizer) after inserting the appropriate THINDBH keywords (from the optimizer's output file) into the keyword file for Prognosis. This procedure is useful for another reason: if the yields from Prognosis match the yields found by the optimizer (as they well should), then everything is working properly, both with the programs and the user.

Following the segment of code that opens the Prognosis files, there is a small block of code that echoes the optimization control information to the output file. This is described in section 3.3.

Harvest controls are initialized in two steps. First, the harvest control for each diameter class, period, and species group is assigned the value that was previously read from the input file. Second, if random harvest controls are desired to begin the optimization, random numbers between 0 and 1 are computed for controls in periods when harvests are scheduled. The user chooses the random number seed in this process. The controls are stored in two arrays, *U1* and *U2*, which are identical at the start. Harvest controls with a value of 0.0 indicate that no harvesting will take place. During the optimization, *U1* is the current optimal regime, and *U2* is a candidate regime obtained with a coordinate search and passed to Prognosis for evaluation.

The present value of the initial regime *U1* is computed with four subroutine calls. First, the Prognosis variables stored on the scratch file are obtained with BHGET. Second, the harvest controls in *U1* are copied onto the proper Prognosis variables using the OPADD routines contained in GKEY. The OPADD routines do the same thing as THINDBH keywords in the keyword file, except that they are dynamic and therefore eliminate the need to predetermine the thinning levels. This feature is clearly the key that allows the optimizer to unlock Prognosis. The arguments for OPADD include the period, the diameter class limits, the harvest control, and the species number. Third, PMAIN2 is called with *JSET*<0 so that the stand is projected for the full number of cycles (NUMCYCLE is specified in the keyword file). When a thinning is implemented in Prognosis, all tree records that fall into the specified diameter and species class have their tree factor (the number of trees per acre represented by the tree record) reduced by the proportion represented by the harvest control. Within PMAIN2, a subroutine called STATE is called in each period. This classifies tree records by diameter class and species group. Before-harvest trees per acre and volume per acre by diameter and species class are stored in arrays *X* and *V*, respectively. Finally, after the stand projection, present value is calculated by calling VALUE. For each merchantable diameter and species class, value is the product of total volume before cut *V*, discounted price per unit volume *P*, and the proportion of trees cut *U*. For each unmerchantable class, value is the product of total number of trees before cut *X*, discounted price per tree *P*, and the proportion of trees cut *U*.

7.2 Search Process

Each iteration of the Method of Hooke and Jeeves involves a set of coordinate searches followed by an evaluation of the termination criteria and one pattern search. A coordinate search involves varying one control variable by a small amount and evaluating the new management regime. If the new management regime has a higher present value, the changed control variable is accepted and the next control variable is varied. The sequence of searches starts with the first cutting period and the smallest diameter class. Note that if the class does not contain trees, no search and evaluation takes place. Searches may go in both positive and negative directions, depending on the initial value of the harvest control. The present value of a candidate regime is computed using the same sequence of subroutine calls (BHGET, GKEY, PMAIN2, and VALUE), with *JSET*<0.

After coordinate searches have been performed for the control variables in the first harvest period, the state variables *X* must be restored to the values associated with the current optimum *U2*. This is done using BHGET, GKEY, and PMAIN2 with *JSET*<0. In addition, the stand is grown to the beginning of the next cutting period and saved. In this block, *JSET* is the number of periods before the second cut. In GKEY, only the harvest controls associated with the current cut are copied to Prognosis variables. PMAIN2 is called to grow the stand, and *JSET* is reset to –1 (in PMAIN2). The Prognosis variables are saved on the scratch file using BHPUT. The coordinate searches for the second growth period now proceed. Note that the evaluation of a control variable change in the second harvest period involves a Prognosis projection that starts with the updated stand at the beginning of that period. This avoids redundant growth projection in periods before the control variable takes effect.

After coordinate searches have been performed for all control variables, the termination criteria are evaluated. If the difference *DIF* in present value between two subsequent management regimes is less than a tolerance *EPS1*, and if the search distance *DELTA* is less than a tolerance *EPS*, then the search process stops and the current regime is printed. Otherwise, a pattern search is conducted.

The first step in the pattern search is to compute the search direction. The direction is *U2-U1*, which is the difference between management regimes before and after the most recent set of coordinate searches. Next, the status of relevant variables in Prognosis must be stored and then reinitialized because we want to start the projection from year 0. (Recall that after the last coordinate search, the stored stand equals the stand just prior to the last cut.) Then, the new management regime is computed by calling CONTROL. The pattern search changes the current regime by moving in the pattern search direction *DIR* by a distance *ALPHA*. The present value of this new regime is computed using BHGET, GKEY, PMAIN2, and VALUE, with *JSET*<0. If the new regime is an improvement, it is saved, the step size for the coordinate search *DELTA* is reduced, and the next set of coordinate searches is started. If the new regime is not better, the old regime is restored, the step size reduced, and the next set of searches started.

8 CREATING AN EXECUTABLE PROGRAM

To give users the option to change the Prognosis Optimization Program to suit their purposes, we have enclosed its source code on the distribution diskette. The source code is long and contains over a hundred subprograms, including all those for the Prognosis Model. Thus, it takes between 1 and 2 hours to create the executable program (most of this time is required to compile the source code). As Wykoff (1988) describes in the appendix of the Prognosis Model Update, there are three steps involved in creating an executable version of the Prognosis Model: (1) compiling the Fortran files, (2) creating object libraries, and (3) linking the object libraries and files. We use the same steps to create the executable version of the Prognosis Optimization Model.

8.1 Compiling Source Code

To compile the Prognosis code, begin by copying files *SOURCE1.ARC*, *SOURCE2.ARC*, and *SOURCE3.ARC* from the program diskette into the directory reserved for the optimizer. *SOURCE1.ARC* and *SOURCE2.ARC* contain the source code for version 5.2 of the Prognosis Model. *SOURCE3.ARC* contains the source code for the optimizer and several utility files for building the executable program. *SOURCE3.ARC* also contains four files that intentionally have the same names as files in

SOURCE1.ARC and *SOURCE2.ARC*: *MAIN.FOR, BLKDAT.FOR, OPFIND.FOR*, and *OPCOM.F77*. These files replace certain common blocks, arrays, and variable declarations that determine the number of keywords that may be used in Prognosis. These limits have been increased to accommodate the large number of THINDBH keywords necessary for optimization. When you extract each of these four files from *SOURCE3.ARC*, the ARC utility will stop and ask if you really want to write over a file that already exists; the correct answer is "yes". To extract all the remaining files, simply execute the *UNLOAD.BAT* file:

UNLOAD

MAIN.FOR contains the MAIN program for the optimizer and replaces the Prognosis main driver *MAIN.FOR* extracted above. The *PMAIN.FOR* file contains subroutines PMAIN1, PMAIN2, and PMAIN3, which together form the Prognosis main driver used in the optimization. Note that almost all of the calls to Prognosis output routines have been disabled in PMAIN3. File *OPT.FOR* contains optimizer subroutines OUT, GKEY, CONTROL, STATE, VALUE, and DRAND. The *BHSTAR.FOR* file contains 12 subroutines adapted from Crookston and Stage (1988) that allow for storing and retrieving the complete status of Prognosis variables and arrays during the course of a projection.

Your optimization directory now contains several hundred Fortran source files (**.FOR*) and common blocks (**.F77*), in addition to the utilities. That number is about to double because compilation will create an **.OBJ* file for each **.FOR* file. We have compiled successfully using Ryan-McFarland version 2.40 (RMFORT). (Wykoff [personal communication 1990] has also produced a working executable module of Prognosis using Lahey F77L version 4.01.) Note that changes in array dimensions contained in *BLKDAT.FOR, OPFIND.FOR*, and *OPCOM.F77* necessitate recompiling the Prognosis routines even if you already have a compiled version of Prognosis. Compile the source code by executing the *COMPILE.BAT* batch file and specifying the appropriate compiler as an argument (we use RMFORT):

COMPILE RMFORT

A relatively long time is needed to compile such a large number of subprograms (approximately 45 minutes on a 20-Mhz personal computer). Note that these programs are all compiled with the /IZ option, which is necessary to make the executable program small enough to run on a microcomputer under the 640K limit imposed by MS-DOS. Also note that the batch commands intentionally do not compile subprograms associated with the COVER extension (Moeur 1985) because they are not needed for unconstrained value optimization. However, an optimization problem that includes constraints on hiding cover for big game, for example, would require them. In that case the only alternatives would be either to recompile Prognosis with smaller dimensions for the tree vectors or to use an operating system such as UNIX that does not have the 640K restriction.

8.2 Building Object Libraries

If you now look at a directory listing, you run the risk of being overwhelmed by the sheer number of files that have been created. Clearly, some sort of file reduction is needed. Object libraries are the answer. By using the Phoenix PLINK86 overlay linker and associated libraries (supplied with the Ryan-McFarland RMFORT compiler), all of the object files needed for the optimizer are loaded into six object libraries: *MAIN.LIB, BASE.LIB, GROWTH.LIB, INPUT.LIB, INPUT2.LIB*, and *ESTAB.LIB*. Execute *LOADPLIB.BAT* to create these libraries:

LOADPLIB ALL

Note that you can also use *LOADPLIB.BAT* to build these libraries one at a time if you replace *ALL* with the appropriate library name in the preceding statement.

All of the source and object files can now be erased. Although this step may seem scary, all of the object files are conveniently stored in one of the six object libraries and all of the source code is still abstracted in the ARC files. Thus, the following commands should considerably reduce bookkeeping problems in your optimization directory:

*ERASE *.OBJ*
*ERASE *.FOR*
*ERASE *.F77*

As strange as it may seem, all of these erased files are still cluttering up your subdirectory, even though they do not appear to be in your directory. They have been marked as deleted in the file allocation table, but the file contents are untouched. The optimizer can be seriously slowed down in such a cluttered directory because it must repeatedly swap overlay segments when it is executed. The simplest solution is to create a new optimization directory, copy everything into it, and delete the old directory. Alternately, you could create a separate run directory containing only the executable optimizer code (including overlays), input and output files, response files, and any batch files that will simplify the submission of runs.

8.3 Linking the Object Code

The size of the optimization code requires the use of program overlays in the executable module. To link the object code and create the overlays, the Phoenix PLINK86 overlay linker is used. Batch file *OPTPLK.BAT* and response file *OPTPLK.RSP* contain the instructions for creating the overlaid version of the executable module. Execute the batch file with the command:

OPTPLK

The resulting executable module, *OPTGROW.EXE*, should require about 529K. The three overlays are *ESESTAB.OVL, ESGROWTH.OVL*, and *ESINPUT.OVL*.

8.4 Modifying the Program

If it is necessary to modify the optimizer code for some reason (such as increasing variable dimensions), then only a few steps are required to produce an executable module. After revising the appropriate source code, compile by specifying the program name and the /IZ option as arguments to the compiler. For example, assume that the subprogram STATE has been changed. Create a new *OPT.OBJ* file (where STATE is stored) by executing the Ryan-McFarland compiler:

RMFORT OPT / IZ

Next, update the appropriate object library (*MAIN.LIB* in this case) by using the *MODLOAD.BAT* batch file:

MODLOAD MAIN OPT

Finally, update the executable module *OPTGROW.EXE* by relinking:

OPTPLK

You will also need to update the ARC source file if you plan to work from the archive in the future:

ARC U SOURCE3 STATE.FOR

The other useful ARC options are D to delete a file and V to get a verbose listing of the contents of the archive file. The syntax is the same as for the X and U options that have been illustrated.

9 REFERENCES

Crookston, N. L. 1985. User's guide to the Event Monitor: an addition to the Prognosis Model. Gen. Tech. Rep. INT-196. Ogden, UT: U.S. Department of Agriculture, Forest Service, Intermountain Research Station. 36 p.

Crookston, N. L.; Stage, A. R. 1988. User's guide to the Parallel Processing Extension of the Prognosis Model. Review draft on file at the Intermountain Research Station, 1221 S. Main, Moscow, ID 83843.

Ferguson, D. E.; Crookston, N. L. 1984. User's guide to the Regeneration Establishment Model—a Prognosis Model extension. Gen. Tech. Rep. INT-161. Ogden, UT: U.S. Department of Agriculture, Forest Service, Intermountain Forest and Range Experiment Station. 23 p.

Ferguson, D. E.; Stage, A. R.; Boyd, R. J. 1986. Predicting regeneration in the grand fir-cedar-hemlock ecosystem of the Northern Rocky Mountains. Forest Science Monograph 26. 41 p.

Haight, R. G.; Monserud, R. A. 1990a. Optimizing any-aged management of mixed-species stands: I. Performance of a coordinate-search process. Canadian Journal of Forest Research. 20: 15-25.

Haight, R. G.; Monserud, R. A. 1990b. Optimizing any-aged management of mixed-species stands: II. Effects of decision criteria. Forest Science. 36(1): 125-144.

Hamilton, D. A., Jr. 1986. A logistic model of mortality in thinned and unthinned mixed conifer stands of northern Idaho. Forest Science. 32: 989-1000.

Hooke, R.; Jeeves, T. A. 1961. Direct search solution of numerical and statistical problems. Journal of the Association for Computing Machinery. 8: 212-221.

Moeur, M. 1985. COVER: a user's guide to the CANOPY and SHRUBS extension of the Stand Prognosis Model. Gen. Tech. Rep. INT-190. Ogden, UT: U.S. Department of Agriculture, Forest Service, Intermountain Research Station. 49 p.

Wykoff, W. R.; Crookston, N. L.; Stage, A. R. 1982. User's guide to the Stand Prognosis Model. Gen. Tech. Rep. INT-133. Ogden, UT: U.S. Department of Agriculture, Forest Service, Intermountain Forest and Range Experiment Station. 112 p.

Wykoff, W. R. 1986. Supplement to the user's guide for the Stand Prognosis Model-Version 5.0. Gen. Tech. Rep. INT-208. Ogden, UT: U.S. Department of Agriculture, Forest Service, Intermountain Research Station. 36 p.

Wykoff, W. R. 1988. Prognosis Model update version 5.2 (Fortran 77). Review draft on file at the Intermountain Research Station, 1221 S. Main, Moscow, ID 83843.

10 APPENDIX: OPTIMIZER PROGRAM CODE

```
      PROGRAM MAIN

C     OPTIMIZE PROGNOSIS USING THE COORDINATE GRID SEARCH METHOD
C     OF HOOK AND JEEVES.

C     WRITTEN BY ROBERT G. HAIGHT 1988 AND MODIFIED IN 1989 BY
C     ROBERT A. MONSERUD  (PROGRAMMING DIVISION OF "THE BOBS")

C     OPTIMIZATION PROBLEM BOUNDS:
C       MAXIMUM NUMBER OF PROGNOSIS GROWTH PERIODS = 33
C       MAXIMUM NUMBER OF DIAMETER CLASSED         =  8
C       MAXIMUM NUMBER OF SPECIES GROUPS           =  3
C     IF THESE DIMENSIONS ARE CHANGED, BE SURE TO MODIFY AND RECOMPILE
C     SUBROUTINES CONTROL, OUT, STATE, VALUE, GKEY, AND PMAIN2.

      COMMON/BEST/DCL(8,2),D(8),U1(33,8,3),U2(33,8,3),DIR(33,8,3),
     >IYEAR(33),X(33,8,3),V(33,8,3),P(33,8,3),BF(33),VA(33),NS(3),
     >SP(3,10),ICUT(33),NCYCLE,NCLASS,NGROUP,MVOL

      DIMENSION C(33,8,3),DCB(9),ISP(10)
      DOUBLE PRECISION DRAND,SEED(20),RAND

      OPEN(25,FILE='BOBS2.INP',STATUS='UNKNOWN',ACCESS='SEQUENTIAL')
      OPEN(26,FILE='BOBS2.OUT',STATUS='UNKNOWN',ACCESS='SEQUENTIAL')
      OPEN(UNIT=50,STATUS='SCRATCH',FORM='UNFORMATTED')

      ISKIP = 0

C************************************************************************
C     BEGIN INPUT AND INITIALIZATION PHASE
C************************************************************************

C*****READ NUMBER OF RUNS AND RANDOM # SEEDS
      READ(25,259) NRUN,(SEED(NR),NR = 1,NRUN)
259   FORMAT(I5,20D5.0)

C*****READ OPTIMIZER CONTROL PARAMETERS
      READ(25,257) R,ALPHA,DELTA1,EPS,EPS1
257   FORMAT(20F5.0)
      MAXVOL=0
      IF(R.LT.0.0001) MAXVOL=1

C*****READ INTERNAL PARAMETERS
C     NCYCLE = 1 + NUMBER OF PROGNOSIS CYCLES                      (TIME)
C     MERCH = INDEX FOR THE FIRST MERCHANTABLE DIAMETER CLASS
C     NGROUP = NUMBER OF SPECIES GROUPS FOR VALUE CLASSIFICATION   (SPP)
C     MVOL > 0 IF BOARD FT VOL; MVOL < 0 IF MERCH CUFT VOL
C     LENGTH = LENGTH OF THE GROWTH PERIOD (TIMEINT)
C     NTH = PRINT EVERY NTH PROGNOSIS CYCLES
C     NOKEYW = 0 IF YOU WANT NO THINDBH KEYWORD OUTPUT WRITTEN

      READ(25,258) NUMCYC,MERCH,NGROUP,MVOL,LENGTH,NTH,NOKEY
258   FORMAT(10I5)
      NCYCLE = NUMCYC + 1
      MERCH1 = MERCH - 1
      YEARS = FLOAT( NUMCYC * LENGTH )
      IF(NTH.EQ.0) NTH = 1

C*****READ CUTTING PERIODS
C     NCUTS = NUMBER OF CUTTING PERIODS
      READ(25,258) NCUTS,(ICUT(I),I = 1,NCUTS)
C     JS = NUMBER OF PERIODS BEFORE THE FIRST CUTTING
      JS = ICUT(1) - 1
      IF(JS.EQ.0) JS = -1
      JSET = JS

C*****READ THE NCLASS+1 DIAMETER CLASS BOUNDARYS
      READ(25,255) NCP1,(DCB(K),K = 1,NCP1)
255   FORMAT(I5,20F5.0)
C     NCLASS = NUMBER OF DIAMETER CLASSES                          (SIZE)
      NCLASS = NCP1 - 1
      DO 15 K = 1,NCLASS
       DCL(K,1) = DCB(K)
       DCL(K,2) = DCB(K+1)
       D(K) =  (DCB(K) + DCB(K+1)) /2.
15    CONTINUE

C*****FOR THE NGROUP SPECIES GROUPS READ THE SPECIES ID NUMBERS
      DO 13 L = 1,NGROUP
        READ(25,255) NSP, (SP(L,L1),L1=1,NSP)
        NS(L) = NSP
13    CONTINUE

C*****READ AND COMPUTE DISCOUNTED STUMPAGE PRICES
      DO 12 L = 1,NGROUP
      READ(25,257) (P(1,J,L),J = 1,NCLASS)
      DO 10 I = 1,NCYCLE
        DIS = (1.+R) ** ((I - 1)*LENGTH)
        DO 11 J = 1,NCLASS
           P(I,J,L) = P(1,J,L)/DIS
11      CONTINUE
10    CONTINUE
12    CONTINUE

C*****READ THINNING CONTROL VARIABLES (PERCENTAGE OF TREES REMOVED)
C     OUTER LOOP FOR SPP GROUPS,
C     INNER LOOP FOR TIME          (DOWN),
C     IMPLIED LOOP FOR DIA CLASSES (ACROSS)
      DO 39 L = 1,NGROUP
       DO 36 I = 1,NUMCYC
        READ (25,256) (C(I,K,L),K = 1,NCLASS)
256     FORMAT(10F7.4)
36     CONTINUE
        DO 38 K = 1,NCLASS
        C(NCYCLE,K,L) = 1.0
38    CONTINUE
39    CONTINUE

C************************************************************************
C     OUTER LOOP FOR SEPERATE OPTIMIZATIONS WITH DIFFERENT SEEDS
C************************************************************************

      DO 999 NR = 1,NRUN

      NCALLS = 0
      DELTA = DELTA1

C*****CALL PROGNOSIS TO OPEN TREELIST FILE 2 AND KEYWORD FILE 5
      REWIND 2
      REWIND 5
      CALL PMAIN1(ISKIP)
      IF (JSET.LT.0) THEN
        CALL BHPUT
      ELSE
        CALL PMAIN2(JSET)
      ENDIF

C*****PRINT INPUT
      WRITE (26,648)
648   FORMAT(///'BOBS2: PROGNOSIS OPTIMIZATION BY THE BOBS')
      SEED1 = SNGL(SEED(NR))
```

```
      WRITE(26,649) NR,R,ALPHA,DELTA,EPS,EPS1,SEED1
649   FORMAT(/,
     >'OPTIMIZATION NUMBER  ',I4/,
     >'INTEREST RATE =      ',F8.3/,
     >'ACCELERATION STEP =  ',F8.3/,
     >'BEGINNING STEP SIZE =',F8.3/,
     >'MINIMUM STEP SIZE =  ',F8.3/,
     >'MINIMUM GAIN =       ',F8.3/,
     >'RANDOM NUMBER SEED = ',F8.3/)

      IF(NCUTS.GT.1) WRITE(26,658) NCUTS,(ICUT(K),K = 1,NCUTS)
658   FORMAT('THE OPTIMIZER WILL DETERMINE',I3,' HARVESTS.'/
     >'THEY OCCUR IN PERIODS:',20I4
     >/22X,13I4)
      IF(NCUTS.EQ.1) WRITE(26,659) ICUT(1)
659   FORMAT('THE OPTIMIZER WILL DETERMINE 1 HARVEST,',
     >' OCCURRING IN PERIOD',I3)
      WRITE(26,651)
651   FORMAT(' ')
      DO 30 L = 1,NGROUP
        WRITE(26,650) L,(SP(L,L1),L1 = 1,NS(L))
650     FORMAT('SPECIES CODES FOR GROUP',I2,' ARE',10F4.0)
30    CONTINUE
      WRITE(26,652)
652   FORMAT(/'HOOKE & JEEVES COORDINATE SEARCH PROCESS: '/)

C***********************************************************************
C     BEGIN SEARCH PROCESS
C***********************************************************************

C*****FOR EACH SPECIES GROUP
      DO 31 L = 1,NGROUP

C*****SET CONTROL VARIABLES
      DO 33 I = 1,NCYCLE
       DO 32 K = 1,NCLASS
         U1(I,K,L) = C(I,K,L)
32     CONTINUE
33    CONTINUE

C*****SET RANDOM STARTING VALUES
      IF(SEED(NR).GE.0.D0) THEN
      DO 26 I1 = 1,NCUTS
        I = ICUT(I1)
        DO 27 K = 1,NCLASS
          RAND = DRAND(SEED(NR))
          U1(I,K,L) = SNGL(RAND)
27      CONTINUE
26    CONTINUE
      ENDIF

C*****SET U2(I,K,L) = U1(I,K,L)
       DO 28 I = 1,NCYCLE
        DO 29 K = 1,NCLASS
          U2(I,K,L) = U1(I,K,L)
          X(I,K,L) = 0.
29      CONTINUE
28     CONTINUE
31    CONTINUE

C*****COMPUTE PRESENT VALUE OF U1
      CALL BHGET
      CALL GKEY(DCL,U1,NCLASS,NGROUP,NS,SP,ICUT,NCUTS,1,LENGTH)
      CALL PMAIN2(JSET)
      CALL VALUE(Z1,P,V,X,U1,NCYCLE,NCLASS,MERCH,NGROUP)
      Z2 = Z1
      NCALLS = NCALLS + 1
      WRITE(26,635) Z1
635   FORMAT('PRESENT VALUE =',F10.2)

C*****FOR EACH CUTTING PERIOD I
49    CONTINUE
      DO 46 I1 = 1,NCUTS
        I = ICUT(I1)

C*****FOR EACH DIAMETER AND SPECIES CLASS J AND L
        DO 47 J = 1,NCLASS
         DO 48 L = 1,NGROUP

C*****ADD DELTA TO TEST CONTROL U2(I,J,L)
          IF(X(I,J,L).LT..01) GOTO 48
          IF(U1(I,J,L).EQ.1.) GOTO 43
          U2(I,J,L) = U1(I,J,L) + DELTA
          IF(U2(I,J,L).GT.1.) U2(I,J,L) = 1.

C*****GET THE INITIAL STAND AND COMPUTE PV OF CHANGED CONTROL
          CALL BHGET
          CALL GKEY(DCL,U2,NCLASS,NGROUP,NS,SP,ICUT,NCUTS,I1,LENGTH)
          CALL PMAIN2(JSET)
          CALL VALUE(Z,P,V,X,U2,NCYCLE,NCLASS,MERCH,NGROUP)
          NCALLS = NCALLS + 1
          IF(Z.GT.Z2) GOTO 40

C*****IF PLUS DELTA DOES NOT IMPROVE PV, SUBTRACT DELTA TO TEST CONTROL
          IF(U1(I,J,L).EQ.0.) GOTO 42
43        CONTINUE
          U2(I,J,L) = U1(I,J,L) - DELTA
          IF(U2(I,J,L).LT.0.) U2(I,J,L) = 0.

C*****GET THE INITIAL STAND AND COMPUTE PV OF CHANGED CONTROL
          CALL BHGET
          CALL GKEY(DCL,U2,NCLASS,NGROUP,NS,SP,ICUT,NCUTS,I1,LENGTH)
          CALL PMAIN2(JSET)
          CALL VALUE(Z,P,V,X,U2,NCYCLE,NCLASS,MERCH,NGROUP)
          NCALLS = NCALLS + 1
          IF (Z.GT.Z2) GOTO 40
42        CONTINUE
          U2(I,J,L) = U1(I,J,L)
          Z = Z2
40        CONTINUE
          Z2 = Z
48       CONTINUE
47      CONTINUE

C*****RECOMPUTE STATE USING BEST CONTROL SEQUENCE
        CALL BHGET
        CALL GKEY(DCL,U2,NCLASS,NGROUP,NS,SP,ICUT,NCUTS,I1,LENGTH)
        CALL PMAIN2(JSET)
        NCALLS = NCALLS + 1

C*****PARK THE STAND AFTER I1 CUTS
        IF(I1.LT.NCUTS) THEN
         JSET = ICUT(I1 + 1) - 1
         CALL BHGET
         CALL GKEY(DCL,U2,NCLASS,NGROUP,NS,SP,ICUT,I1,I1,LENGTH)
         CALL PMAIN2(JSET)
         NCALLS = NCALLS + 1
        ENDIF
46    CONTINUE

C*****NOTE TERMINATION OF STEP SEARCH LOOP
      WRITE(26,637) DELTA
```

```
637   FORMAT(' PERFORMED STEP SEARCH WITH DELTA =  ',F5.3)
      WRITE(26,635) Z2

C*****IF DIF .LT. EPS1 .AND. DELTA. .LT. EPS, GO TO PRINT
        DIF = Z2 - Z1
        IF(DIF.LT.EPS1.AND.DELTA.LT.EPS) GOTO 90

C*****PERFORM ACCELERATION STEP: FIRST DETERMINE SEARCH DIRECTION
        DO 50 I1 = 1,NCUTS
         I = ICUT(I1)
         DO 52 L = 1,NGROUP
          DO 51 J = 1,NCLASS
            DIR(I,J,L) = U2(I,J,L) - U1(I,J,L)
            U1(I,J,L) = U2(I,J,L)
51        CONTINUE
52       CONTINUE
50      CONTINUE

C*****NEXT, REINITIALIZE AND PUT THE STAND
      REWIND 2
      REWIND 5
      JSET = JS
      CALL PMAIN1(ISKIP)
      IF(JSET.LT.0) THEN
        CALL BHPUT
      ELSE
        CALL PMAIN2(JSET)
      ENDIF

C*****NEXT, COMPUTE PRESENT VALUE
          Z1 = Z2
          CALL CONTROL(ALPHA,U2,DIR,NCLASS,NGROUP,ICUT,NCUTS)
          CALL BHGET
          CALL GKEY(DCL,U2,NCLASS,NGROUP,NS,SP,ICUT,NCUTS,1,LENGTH)
          CALL PMAIN2(JSET)
          CALL VALUE(Z2,P,V,X,U2,NCYCLE,NCLASS,MERCH,NGROUP)
        NCALLS = NCALLS + 1

C*****IF IMPROVEMENT, SAVE AND RETURN TO COORDINATE SEARCH
        IF(Z2.GT.Z1) THEN
        WRITE(26,636)
636     FORMAT('  ACCELERATION STEP SUCCESSFUL')
        WRITE(26,635) Z2
        DO 54 I1 = 1,NCUTS
         I = ICUT(I1)
         DO 57 L = 1,NGROUP
          DO 55 J = 1,NCLASS
           U1(I,J,L) = U2(I,J,L)
55        CONTINUE
57       CONTINUE
54      CONTINUE
        Z1 = Z2
        IF(DELTA.GT.EPS) DELTA = DELTA/2.
        GOTO 49

C*****IF NO IMPROVEMENT, REDUCE STEP SIZE AND RETURN TO COORDINATE
        ELSE
        WRITE(26,656)
656     FORMAT('  ACCELERATION STEP NOT SUCCESSFUL')
        DO 70 I1 = 1,NCUTS
         I = ICUT(I1)
         DO 72 L = 1,NGROUP
          DO 71 J = 1,NCLASS
           U2(I,J,L) = U1(I,J,L)
71        CONTINUE
72       CONTINUE
70      CONTINUE
        Z2 = Z1
        IF(DELTA.GT.EPS) DELTA = DELTA/2.
        GOTO 49
        ENDIF

90    CONTINUE

C***********************************************************************
C     SEARCH PROCESS COMPLETE; PRINT RESULTS
C***********************************************************************

C*****COMPUTE AND SUBTRACT INITIAL VALUE (OR VOLUME)

      VAINIT = 0.
      DO 80 L = 1,NGROUP
        DO 86 J = 1,MERCH1
          VAINIT = VAINIT + P(1,J,L) * X(1,J,L)
86      CONTINUE
        DO 89 J = MERCH,NCLASS
          VAINIT = VAINIT + P(1,J,L) * V(1,J,L)
89      CONTINUE
80    CONTINUE
      Z2NET = Z2 - VAINIT

C*****PRINT OPTIMAL VALUE

      IF(MAXVOL.EQ.0) WRITE(26,626) NCALLS,VAINIT,Z2NET
626   FORMAT(/'NUMBER OF PROGNOSIS SIMULATIONS =',I5//
     >'INITIAL VALUE = ',F10.2,' DOLLARS/AC'//
     >'OPTIMAL PRESENT NET VALUE (PNV) =',F10.2,' DOLLARS/AC'/ )

      VAINIT = VAINIT * 1000.
      Z2NET = Z2NET * 1000.
      AAP = Z2NET / YEARS

      IF(MAXVOL.EQ.1.AND.MVOL.GT.0)WRITE(26,627) NCALLS,VAINIT,Z2NET,AAP
627   FORMAT(/'NUMBER OF PROGNOSIS SIMULATIONS =',I5//
     >'INITIAL VOLUME =    ',F14.1,' BDFT/AC'//
     >'NET OPTIMAL VOLUME =',F14.1,' BDFT/AC'//
     >'AVERAGE ANNUAL PRODUCTION =',F7.1,' BDFT/AC/YR'/)

      IF(MAXVOL.EQ.1.AND.MVOL.LE.0)WRITE(26,628) NCALLS,VAINIT,Z2NET,AAP
628   FORMAT(/'NUMBER OF PROGNOSIS SIMULATIONS =',I5//
     >'INITIAL VOLUME =    ',F14.1,' CUFT/AC'//
     >'NET OPTIMAL VOLUME =',F14.1,' CUFT/AC'//
     >'AVERAGE ANNUAL PRODUCTION =',F7.1,' CUFT/AC/YR'/)

C*****COMPUTE OUTPUT YEAR ARRAY
      DO 66 I = 1,NCYCLE
        IYEAR(I) = (I - 1) * LENGTH
66    CONTINUE

C*****WRITE RESIDUAL STATE SEQUENCE
      DO 60 L = 1,NGROUP
      DO 58 I = 1,NCYCLE
        VA(I) = 0.
        BF(I) = 0.
        DO 56 J = 1,MERCH1
          U1(I,J,L) = X(I,J,L) * (1. - U2(I,J,L))
          VA(I) = VA(I) + P(1,J,L) * U1(I,J,L)
56      CONTINUE
        DO 59 J = MERCH,NCLASS
          U1(I,J,L) = X(I,J,L) * (1. - U2(I,J,L))
          VA(I) = VA(I) + P(1,J,L) * V(I,J,L) * (1. - U2(I,J,L))
          BF(I) = BF(I) + V(I,J,L) * (1. - U2(I,J,L))
```

```
59      CONTINUE
58    CONTINUE
      WRITE(26,657) L
657   FORMAT(/,'RESIDUAL TREES PER ACRE FOR SPECIES GROUP',I3)
      CALL OUT(IYEAR,D,U1,BF,VA,NCYCLE,NCLASS,L,0,NTH)
60    CONTINUE

C*****WRITE HARVEST SEQUENCE
      DO 68 L = 1,NGROUP
      DO 61 I = 1,NCYCLE
        VA(I) = 0.
        BF(I) = 0.
        DO 67 J = 1,MERCH1
          U1(I,J,L) = X(I,J,L) * U2(I,J,L)
          VA(I) = VA(I) + P(1,J,L) * U1(I,J,L)
67      CONTINUE
        DO 62 J = MERCH,NCLASS
          U1(I,J,L) = X(I,J,L) * U2(I,J,L)
          VA(I) = VA(I) + P(1,J,L) * V(I,J,L) * U2(I,J,L)
          BF(I) = BF(I) + V(I,J,L) * U2(I,J,L)
62      CONTINUE
61    CONTINUE
      WRITE(26,660) L
660   FORMAT(/,'HARVESTED TREES PER ACRE FOR SPECIES GROUP',I3,':')
      CALL OUT(IYEAR,D,U1,BF,VA,NCYCLE,NCLASS,L,0,NTH)
68    CONTINUE

C*****WRITE CONTROL SEQUENCE
      DO 69 L = 1,NGROUP
      DO 63 I = 1,NCYCLE
        DO 64 J = 1,NCLASS
        IF(U1(I,J,L).GT.0.001)  U1(I,J,L)  =  U2(I,J,L) * 100.
64      CONTINUE
63    CONTINUE
      WRITE(26,633) L
633   FORMAT(/,'PERCENTAGE TREES PER ACRE CUT FOR SPECIES GROUP'
     >,I3,':')
      CALL OUT(IYEAR,D,U1,BF,VA,NCYCLE,NCLASS,L,1,NTH)
69    CONTINUE


C*****REWRITE CONTROL SEQUENCE IN THE INPUT FORMAT
C     (DIAMETER CLASSES ACROSS, TIME PERIODS DOWN)
      DO 85 L = 1,NGROUP
      DO 83 I = 1,NCYCLE
        DO 84 J = 1,NCLASS
          U1(I,J,L)  =  U1(I,J,L) / 100.
84      CONTINUE
83    CONTINUE

      WRITE(26,670) L
670   FORMAT(/,'OPTIMAL HARVEST CONTROL PARAMETERS FOR SPECIES GROUP'
     >,I3,':')
        DO 87 I = 1,NUMCYC
        WRITE(26,256) (U1(I,J,L),J = 1,NCLASS)
87    CONTINUE

85    CONTINUE

      IF(NOKEY.EQ.0) GOTO 990
C*****WRITE CONTROL SEQUENCE AS THINDBH KEYWORDS FOR PROGNOSIS
      WRITE(26,690)
690   FORMAT(/,'OPTIMAL HARVEST KEYWORDS FOR PROGNOSIS'/
     >' (INSERT IN THE KEYWORD FILE)' )

      DO 94 I = 1,NCUTS
       K = ICUT(I)
       YEAR = (LENGTH * (K - 1)) + 1
        DO 93 J = 1,NCLASS
         J1 = J + 1
         DO 92 L = 1,NGROUP
          IF(U1(K,J,L).LT.0.0001) GOTO 92
          NSP= NS(L)
           DO 91 L1=1,NSP
            SPECIE = SP(L,L1)
            IF(NGROUP.GT.1) THEN
             WRITE(26,696) YEAR,DCB(J),DCB(J1),U1(K,J,L),SPECIE
696          FORMAT('THINDBH   ',3F10.0,F10.4,F10.0 )
            ELSE
             WRITE(26,696) YEAR,DCB(J),DCB(J1),U1(K,J,L)
            ENDIF
91         CONTINUE
92       CONTINUE
93      CONTINUE
94    CONTINUE

990   CONTINUE
999   CONTINUE
      ENDFILE 26
      END

C*************************************************************************
C*************************************************************************

      SUBROUTINE OUT(IYEAR,D,U,BF,VA,NCYCLE,NCLASS,L,L3,NTH)
      DIMENSION IYEAR(33),D(8),U(33,8,3),BF(33),VA(33),TR(33),BA(33)

C*****COMPUTE STAND AVERAGES
      C = 0.00545415
      DO 10 I = 1,NCYCLE
        TR(I) = 0.
        BA(I) = 0.
        DO 11 J = 1,NCLASS
          TR(I) = TR(I) + U(I,J,L)
          BA(I) = BA(I) + U(I,J,L) * C * D(J)**2
11      CONTINUE
10    CONTINUE

C*****WRITE STAND STATISTICS

      N1 = 1
      N2 = NCYCLE
      NPAGE = 1
      N3=(NCYCLE-1)/NTH  + 1
      IF(N3.GT.17) NPAGE = 2
      IF(NPAGE.GT.1) N2 = 17 * NTH

      DO 100 IPAGE = 1,NPAGE
      IF(IPAGE.EQ.2) N1 = N2 + 1
      IF(IPAGE.EQ.2) N2 = NCYCLE

      WRITE(26,401)
401     FORMAT(/,' DBH     --------YEAR------->')
      WRITE(26,402) (IYEAR(I),I = N1,N2,NTH )
402     FORMAT('CLASS',1X,17(2X,I3,2X))
      IF(L3.GT.0) GOTO 80
      DO 40 J = 1,NCLASS
        WRITE(26,403) D(J),(U(I,J,L),I = N1,N2,NTH)
403     FORMAT(F4.0,1X,17F7.1)
40    CONTINUE
```

```
      WRITE(26,412) (TR(I),I = N1,N2,NTH)
412    FORMAT('TOTAL',17F7.0)
      WRITE(26,413) (BA(I),I = N1,N2,NTH)
413    FORMAT('BA/AC',17F7.0)
      WRITE(26,407) (BF(I),I = N1,N2,NTH)
407     FORMAT('VO/AC',17F7.2)
      WRITE(26,414) (VA(I),I = N1,N2,NTH)
414    FORMAT('$$/AC',17F7.1)
      GOTO 100

80    CONTINUE
      DO 90 J = 1,NCLASS
        WRITE(26,404) D(J),(U(I,J,L),I = N1,N2,NTH)
404     FORMAT(F4.0,1X,17F7.2)
90    CONTINUE

100   CONTINUE
      RETURN
      END

C*****************************************************************************
C*****************************************************************************

      SUBROUTINE CONTROL(THETA,U2,DIR,NCLASS,NGROUP,ICUT,NCUTS)
      DIMENSION U2(33,8,3),DIR(33,8,3),ICUT(33)


      DO 15 I1 = 1,NCUTS
       I = ICUT(I1)
       DO 14 L = 1,NGROUP
        DO 16 K = 1,NCLASS
          U2(I,K,L) = U2(I,K,L) + THETA * DIR(I,K,L)
          IF(U2(I,K,L).LT.0.) U2(I,K,L) = 0.
          IF(U2(I,K,L).GT.1.) U2(I,K,L) = 1.
16      CONTINUE
14     CONTINUE
15    CONTINUE
      RETURN
      END

C*****************************************************************************
C*****************************************************************************

      SUBROUTINE VALUE(Z,P,V,X,U,NCYCLE,NCLASS,MERCH,NGROUP)
      DIMENSION P(33,8,3),V(33,8,3),X(33,8,3),U(33,8,3)

C*****COMPUTE PRECOMMERCIAL AND COMMERCIAL THINNING REVENUES
C*****FOR EACH TIME PERIOD

      MERCH1=MERCH-1
      Z=0.
      DO 21 I=1,NCYCLE
       DO 24 L=1,NGROUP
        DO 22 J=1,MERCH1
          Z=Z+P(I,J,L)*X(I,J,L)*U(I,J,L)
22      CONTINUE
        DO 23 J=MERCH,NCLASS
          Z=Z+P(I,J,L)*V(I,J,L)*U(I,J,L)
23      CONTINUE
24     CONTINUE
21    CONTINUE
      RETURN
      END

C*****************************************************************************
C*****************************************************************************

      SUBROUTINE GKEY(DCL,U,NCLASS,NGROUP,NS,SP,ICUTS,NCUTS,NC1,LENGTH)
      DIMENSION DCL(8,2),U(33,8,3),NS(3),SP(3,10),ICUTS(33),
     >WK6(4)

C*****REWIND PROGNOSIS INPUT AND OUTPUT FILES
C*****TREE DATA FILE
      REWIND 2
C*****TREE LIST OUTPUT FILE
      REWIND 3
C*****SUMMARY OUTPUT FILE
      REWIND 4
C*****OUTPUT FILE
      REWIND 6

C*****FOR EACH CUT CYCLE WRITE THINDBH KEYWORDS USING OPADD ROUTINE
      DO 40 I1=NC1,NCUTS
        I=ICUTS(I1)
        IYR=(I-1)*LENGTH+1
        DO 30 L=1,NGROUP
          NSP=NS(L)
          DO 20 L1=1,NSP
            DO 10 K=1,NCLASS
              WK6(1)=DCL(K,1)
              WK6(2)=DCL(K,2)
              WK6(3)=U(I,K,L)
              WK6(4)=SP(L,L1)
              CALL OPADD(IYR,228,4,WK6,KODE)
              IF(KODE.GT.0) STOP 'OPTION PROCESSING ERROR IN GKEY'
10          CONTINUE
20        CONTINUE
30      CONTINUE
40    CONTINUE

      RETURN
      END

C*****************************************************************************
C*****************************************************************************

      SUBROUTINE STATE

      COMMON/BEST/DCL(8,2),D(8),U1(33,8,3),U2(33,8,3),DIR(33,8,3),
     >IYEAR(33),X(33,8,3),V(33,8,3),P(33,8,3),BF(33),VA(33),NS(3),
     >SP(3,10),ICUT(33),NCYCLE,NCLASS,NGROUP,MVOL

      INCLUDE 'ARRAYS.F77'
      INCLUDE 'CONTRL.F77'
      INCLUDE 'PLOT.F77'

      I=ICYC+1

C*****SET X(I,J,L)=0.
      DO 10 L=1,NGROUP
        DO 5 J=1,NCLASS
          X(I,J,L)=0.
          V(I,J,L)=0.
5       CONTINUE
10    CONTINUE
```

```
C*****FOR EACH RECORD READ SPECIES, TPA, DBH AND VOLUME
      DO 40 J1=1,ITRN
        SP1=ISP(J1)
        TPA=PROB(J1)/GROSPC
        BDBH=DBH(J1)
        IF(MVOL.GT.0) THEN
          TVOL=BFV(J1)
        ELSE
          TVOL=WK1(J1)
        ENDIF
        IF(SP(1,1).EQ.0.) SP1=0.

C*****DETERMINE DIAMETER CLASS
        DO 30 J=1,NCLASS
         IF(BDBH.LT.DCL(J,2)) THEN

C*****DETERMINE SPECIES CLASS AND UPDATE TPA AND VOLUME
           DO 22 L=1,NGROUP
            NSPB=NS(L)
            DO 20 L1=1,NSPB
             IF(SP1.EQ.SP(L,L1)) THEN
              X(I,J,L)=X(I,J,L)+TPA
              V(I,J,L)=V(I,J,L)+TVOL*TPA/1000.
              GOTO 40
             ENDIF
20          CONTINUE
22         CONTINUE
         ENDIF
30      CONTINUE
40    CONTINUE
      RETURN
      END
```

```
C***************************************************************************
C***************************************************************************

      DOUBLE PRECISION FUNCTION DRAND(IX)
      DOUBLE PRECISION A,P,IX,B15,B16,XHI,XALO,LEFTLO,FHI,K
      DATA A/16807.D0/,B15/32768.D0/,B16/65536.D0/,P/2147483647.D0/
      XHI = IX/B16
      XHI = XHI - DMOD(XHI,1.D0)
      XALO = (IX - XHI*B16)*A
      LEFTLO = XALO/B16
      LEFTLO = LEFTLO - DMOD(LEFTLO,1.D0)
      FHI = XHI*A + LEFTLO
      K = FHI/B15
      K = K - DMOD(K,1.D0)
      IX = (((XALO - LEFTLO*B16) - P) + (FHI - K*B15)*B16) + K
      IF(IX.LT.0.D0) IX = IX + P
      DRAND = IX*4.656612875D - 10
      RETURN
      END
```

Monserud, Robert A.; Haight, Robert G. 1990. A programmer's guide to the Progno Optimization Model. Gen. Tech. Rep. INT-269. Ogden, UT: U.S. Department of Agriculture, Forest Service, Intermountain Research Station. 24 p.

The Prognosis Optimization Model is used to compute optimal harvest regimes for mixed-conifer stands in the Northern Rocky Mountains. The optimizer is capable of determining optimal regimes (termed any-aged) without any constraints on the age structure or species composition of the stand. It also may be used to determine optimal thinning regimes for plantations or natural stands that are eventually clearcut. The input requirements and output for the optimization model are described in detail. Input and output for three optimization runs for maximizing the present value and merchantable cubic foot volume of a stand are described. Instructions for installing and running the program on a personal computer are provided. For those interested in changing the source code for the optimizer, the structure of the optimization code is described, and detailed instructions for compiling the program are provided.

KEYWORDS: optimal harvesting, any-aged management, single-tree simulator

INTERMOUNTAIN RESEARCH STATION

The Intermountain Research Station provides scientific knowledge and technology to improve management, protection, and use of the forests and rangelands of the Intermountain West. Research is designed to meet the needs of National Forest managers, Federal and State agencies, industry, academic institutions, public and private organizations, and individuals. Results of research are made available through publications, symposia, workshops, training sessions, and personal contacts.

The Intermountain Research Station territory includes Montana, Idaho, Utah, Nevada, and western Wyoming. Eighty-five percent of the lands in the Station area, about 231 million acres, are classified as forest or rangeland. They include grasslands, deserts, shrublands, alpine areas, and forests. They provide fiber for forest industries, minerals and fossil fuels for energy and industrial development, water for domestic and industrial consumption, forage for livestock and wildlife, and recreation opportunities for millions of visitors.

Several Station units conduct research in additional western States, or have missions that are national or international in scope.

Station laboratories are located in:

Boise, Idaho

Bozeman, Montana (in cooperation with Montana State University)

Logan, Utah (in cooperation with Utah State University)

Missoula, Montana (in cooperation with the University of Montana)

Moscow, Idaho (in cooperation with the University of Idaho)

Ogden, Utah

Provo, Utah (in cooperation with Brigham Young University)

Reno, Nevada (in cooperation with the University of Nevada)