

FOLPI, A Forestry-Oriented Linear Programming Interpreter *

Oscar García

Forest Research Institute
New Zealand Forest Service

Summary

FOLPI is a Linear Programming-based planning system that can be used by forest managers without a detailed knowledge of Linear Programming (LP). It is compatible with IFS, an interactive forest simulator widely used by the New Zealand Forest Service. The system allows the user to state the problem in forestry terms, within a framework similar to that used by IFS. The problem statement is automatically translated into the format required by an LP package, using the same input data files as IFS, and an optimal solution is found. The LP output is then interpreted in terms of the original problem and reports identical to those produced by IFS can be generated interactively. The simulator can be used to explore the effect of deviations from the “optimal” solution.

Keywords: Forest management, planning, simulation, Operations Research.

Introduction

Simulation and Linear Programming (LP) are two commonly used tools in forest management planning. Because of their characteristics, they should be viewed as complementary rather than competitive approaches (García

*Pp. 293–305 in: Nagumo, H. *et al.* (eds.) *Proceedings IUFRO Symposium on Forest Management Planning and Managerial Economics*. University of Tokyo. 1984.

1981). LP models force a considerable simplification of management objectives and constraints, and many relevant factors that are difficult to quantify or not easily represented within the LP framework are excluded. Simulation, on the other hand, generally can evaluate only a small proportion of all the possible management alternatives, with the consequent danger of overlooking the most profitable one. With most known implementations it is not easy to alternate between formulations based on LP and simulation because of different conceptual models of the forest and different data requirements. This is particularly true with what Johnson and Scheurman (1977) call Model I, which forms the basis of most of the current LP-based forestry planning models.

Voluntary acceptance of LP by forest managers has also been delayed by the high level of skill required to understand and use these models. Many depend on specialists to formulate the problem, run the system, and interpret the “answers”. This contributes to a credibility gap. There is a need for systems that forest managers can use themselves, without a detailed knowledge of irrelevant mathematical and computational technicalities.

FOLPI is an LP-based forestry planning system which attempts to overcome some of the problems discussed above. The planning problem is stated in forestry terms, automatically translated into a Linear Programming formulation, solved with a standard LP package, and the solution is interpreted in terms of the original problem. The input and output are compatible with IFS, a simple forest simulator widely used by the New Zealand Forest Service (García 1981).

Forest model

The representation of the forest in IFS and FOLPI is similar to some long used by foresters in the manual preparation of cutting schedules. The forest is described by a classification of areas into “crop types” and age classes. Stands are allocated to crop types depending on their growth, silvicultural regime, harvesting methods, location, ownership, or other characteristics, as appropriate to the planning exercise. Events are recorded for time intervals (periods) of length equal to the number of years in each age class.

Figure 1 illustrates the conceptual model for one crop type in an example with 5-year age classes. At the beginning of each period the state of the

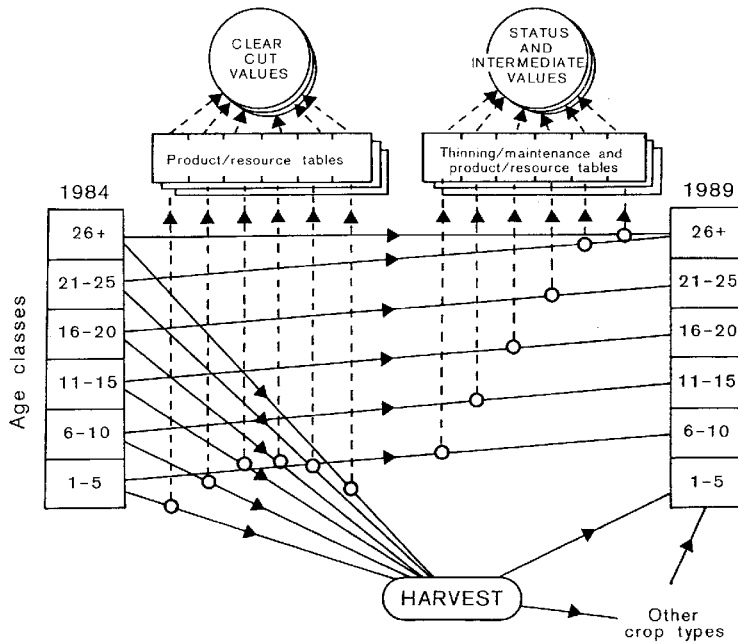


Figure 1:

forest is described by the area in each crop type and age class. Some of the area in each class may be cut, and the remaining area moves into the next age class for the next period. The harvested areas may be replanted immediately into the same or different crop types, or left unplanted. The areas cut from each class are multiplied by the appropriate entries in harvest product/resource tables in order to compute volumes produced or resources required or generated in the period. The residual areas may be multiplied by entries in the same product/resource tables to assess the growing stock, and by entries in another set of tables (here called thinning/maintenance tables) to account for intermediate products/resources such as thinning yields and silvicultural costs. It is also possible to transfer areas between crop types; this is useful for modelling alternative silvicultural regimes or changes of ownership.

The basic data for both IFS and FOLPI consist of tables of initial areas by crop type and age class, and the product/resource and thinning/maintenance tables. The current versions also include a special product/resource table, the “yields”, which is used in conjunction with some of the IFS cutting commands (see García 1981). The yields are usually the total

volume per hectare, and the product/resource and thinning/maintenance tables may represent volumes or values for various products, stumpages, costs, or resources such as man-hours or machinery.

In IFS the areas to be cut or planted are simulated interactively. A number of commands are available to specify these decisions, to produce reports, and to advance or backtrack the simulation. FOLPI, by contrast, requires the user to specify a set of constraints, and an objective function to be maximised or minimised over a given number of periods. The specification of the objective and constraints is described in the following section.

Some enhancements have been incorporated into IFS since it was described in García (1981), including an increase in the allowed number of products/resources, and the computation of discounted and accumulated values.

Problem formulation

The formulation of a problem in FOLPI is best introduced through a very simple (and unrealistic) example. Assume three crop types, named GOOD, AVERAGE and POOR, and seven 5-year age classes. There are three final products: SAWLOGS, PEELERS and PULPLOGS. An example of input for FOLPI, including appropriate comments, follows. Anything to the right of a single quote is a comment, and is ignored by the system. Ages designate the age class that contains them, and crop and product names can be abbreviated.

```
6 ' Number of periods in the optimisation.
'Data file names (contents as in the example in Garcia 1981):
IFS.TESTAREAS
IFS.TESTYIELDS
IFS.TESTPRODS
IFS.TESTTHINNINGS
$ MINIMUM AGE OFF CLEARFELLING
'In any period in crop GOOD, do not cut below age 15:
,GOOD, 15
'In crops AVERAGE to POOR do not cut below age 8:
,AVE - POOR, 8
$ MAXIMUM AGE OF C.F.
' Unspecified.
```

```

$ REPLANTING ' Allowed re lanting, in addition to replanting
' into the same crop type.
' In periods 3 to 6, allow areas cut in crop type POOR
' to be replanted into crop AVERAGE:
3 - 6, POOR, AVERAGE
$ TRANSFERS ' Allowed transfers between crop types.
' None.
$ OBJECTIVE FUNCTION
' Total clearcutting production of SAWLOGS, plus 1.5 times
' the production of PEELERS from crop type GOOD, minus 0.2
' times the total hectares planted:
CUT SAWLOGS (,,) + 1.5 CUT PEEL (,GOOD,) - 0.2 PLANT (,,)
$ CONSTRAINTS
' Residual area in ages 10 to 30 in the last period greater
' than or equal to that in the first:
RES AREA (6,,10-30) - RES AREA (1,,10-30) > 0
' Volume cut in the last 3 periods greater than or equal
' to 15 million cubic metres:
CUT YIELD (4-6,,) > 15000000
$ END

```

The general format of the problem formulation in the current version of FOLPI is as follows ([] indicates optional items, { } indicates alternatives).

```

Number of periods
Area file name
Yields file name
Products/resources file name
[Thinnings/maintenance file name] (optional)
$[MINIMUM AGE OF CLEARFELLING] (optional text)
    period range, crop range, age (any number,
    . see below for
    . definitions)
    .
$[MAXIMUM AGE OF CLEARFELLING]
    period range, crop range, age
    .
    .

```

```

.
$[REPLANTING]
  period range, crop range, crop range
.
.
.
$[TRANSFERS]
  crop range, age range, crop range           (allowed crop
.                                               transfers at time 0,
.                                               bidirectional)
.
$[OBJECTIVE FUNCTION]
  { + } item ± item ± ...                       (may be continued on
  { - }                                         any number of lines)
  { null }
$[CONSTRAINTS]
  { + } item ± item ± ... ± item { < }
  { - }                                         = } constant
  { null }                                       > }
.
.
.
$[END]

```

Ranges may be of any of the three forms:

value1 - value2 specifies all values (periods,
ages or crop types) between
value1 and value2

value specifies the value indicated

null (empty or blank) specifies all possible values.

“Items” in the objective function and constraints are of the form:

$$[\text{constant}] \left\{ \begin{array}{l} \left\{ \begin{array}{l} \text{C[UT]} \\ \text{R[ES]} \\ \text{T[HIN]} \end{array} \right\} \left\{ \begin{array}{l} \text{name} \\ \text{Y[IELD]} \\ \text{A[REA]} \end{array} \right\} \text{ (period range, crop range, age range)} \\ \text{P[LANT]} \text{ (period range, crop range, crop range)} \\ \text{U[SER]} \text{ number} \end{array} \right\}$$

USER items represent auxiliary variables introduced by the user. For example, we are currently experimenting with problem formulations that include decisions concerning utilisation and transport of forest products, where USER variables represent capacity expansions and allocations of products to mills.

The MINIMUM AGE OF C.F. and MAXIMUM AGE OF C.F. sections are useful in reducing the size of the resultant Linear Programming problem. It is assumed that all stands that reach the maximum age of clearfelling are immediately cut.

Planting of new land and removals of land from forestry production are handled through a dummy crop type named BARELAND, which can be used in PLANT items and in the REPLANTING section. BARELAND is not included in crop ranges implied by null fields.

Optimisation

FOLPI automatically translates the problem formulation above into a Linear Programming problem, solves it, and then translates the solution back into easily understandable terms. A user, therefore, does not need to be concerned with the details described in this section

The LP formulation in FOLPI is based on the flow of areas between periods, crop types and age classes depicted in Fig. 1. There are two types of constraints: those supplied by the user in the problem formulation, and the “structural” constraints enforcing conservation of area. The structural constraints are generated by the system and transparent to the user. The user-supplied constraints are transformed by the system to a format acceptable to the LP package and added to the structural constraints.

Figure 2 is derived directly from Figure 1, and shows the flow of areas for one crop type in an example with 4 periods and starting with 4 age classes.

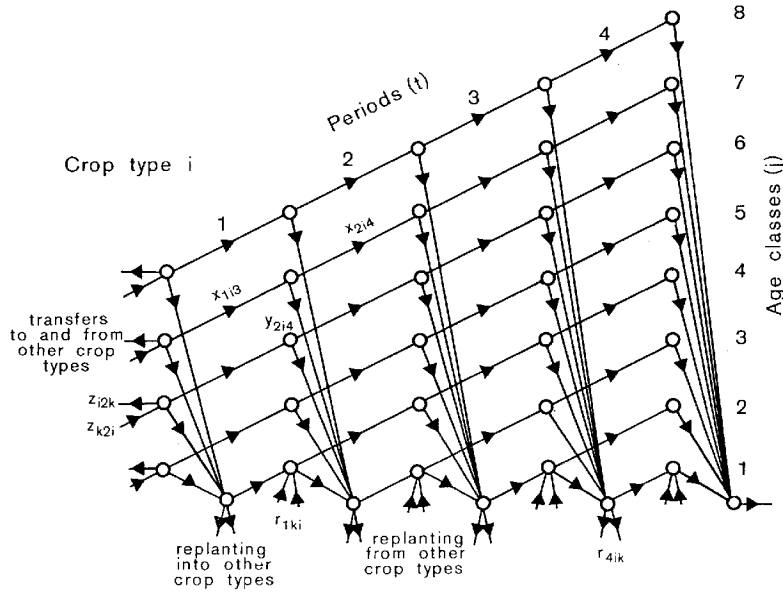


Figure 2:

It is mathematically convenient to add age classes as needed, instead of having an open-ended oldest age class as in Figure 1.

This network could be used directly for an LP formulation. There are four groups of decision variables: The y_{tij} specify the area cut in period t , crop i and age j . The x_{tij} are the residual areas after cutting, which move into the next age class in the next period. The area harvested in period t from crop i and immediately replanted into crop k is denoted by r_{tik} . The z_{ijk} represent possible transfers of area from crop type i and age class j to crop type k . In IFS these transfers can be made in any period, but here it is sufficient to allow transfers only at time zero. In Fig. 2 we have arbitrarily cut everything after the last period T , or, equivalently, renamed the $x_{T+1,i,j}$ variables as $y_{T+1,i,j}$.

The structural constraints simply express the conservation of flow at all the nodes of the network. The CUT items in the user-supplied constraints correspond to linear combinations of y -variables, the RES and THIN items to linear combinations of x -variables, and the PLANT items to sums of r -variables. The REPLANTING and TRANSFERS sections specify which r - and z -variables are allowed to be different from zero.

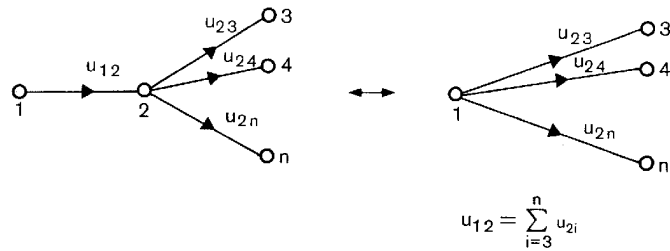


Figure 3:

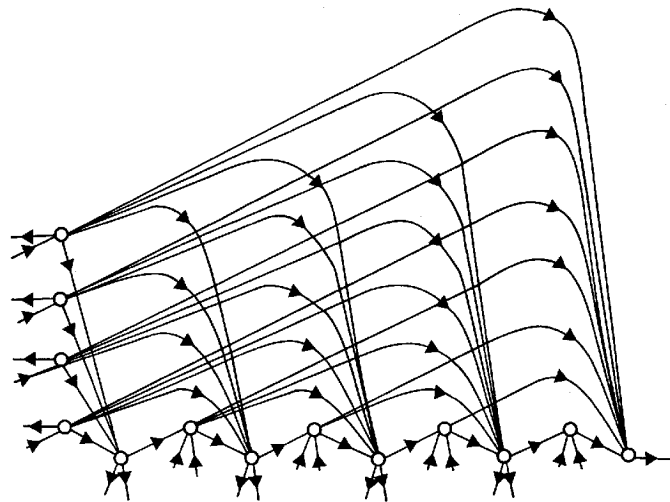


Figure 4:

It is possible, however, to reduce the number of variables and constraints resulting from the network in Fig. 1. We observe that in any network, when there is only one arc entering or leaving a node then that node and arc can be eliminated without loss of information and without relaxing the constraints (Fig. 3). Applying this simplification repeatedly to Fig. 2 results in Fig. 4, where most of the nodes (constraints) and all the x -variables have been eliminated. In addition, a substantial number of the y 's can usually be eliminated using the information in the MINIMUM AGE OF C.F. and MAXIMUM AGE OF C.F. sections.

From the conservation of flow in each of the three kinds of nodes in

Fig. 4, the structural constraints can be written as

$$\sum_j y_{tij} = \sum_k r_{tik} \quad ; t = 1, \dots, T; i = 1, \dots, I \quad (1)$$

$$\sum_k r_{tki} = \sum_{s=t+1}^{T+1} y_{s,i,s-t} \quad ; t = 1, \dots, T; i = 1, \dots, I \quad (2)$$

$$a_{ij} + \sum_k z_{kji} - \sum_k z_{ijk} = \sum_{s=1}^{T+1} y_{s,i,j+s-1} \quad ; i = 1, \dots, I; j = 1, \dots, J \quad (3)$$

$$y_{tij}, r_{tik}, z_{ijk} \geq 0 .$$

Here a_{ij} is the initial area in crop i and age class j , I is the number of crop types, and J is the oldest initial age class. The variables included in these constraints are determined by the **MINIMUM AGE OF C.F.**, **MAXIMUM AGE OF C.F.**, **REPLANTING**, and **TRANSFERS** sections. The z 's may alternatively be considered as unrestricted in sign and defined only for $i > k$. The x 's in the user's constraints are changed to y 's using

$$x_{tij} = \sum_{s=t+1}^{T+1} y_{s,i,j+s-t} . \quad (4)$$

Figure 4 may also be interpreted by designating the flow through the arcs as $w_{ipq} = y_{q,i,q-p}$. The decision variables w_{ipq} are then the area of crop type 1 planted in period p that is cut in period q . An LP formulation along these lines is outlined in Bourgau and Poupardin (1972).

FOLPI solves the problem described above using a standard Linear Programming package. A solution making use of the generalised upper bounds (GUBs) feature available in some LP packages has also been implemented. Changing the sign of the r 's, it is seen that the constraints (2) and those constraints (3) where the first sum does not appear constitute a set of GUBs. Then only the constraints (1), some of those in (3), and the user-supplied constraints need to appear explicitly in the LP matrix. Computational results and other possible solution approaches are discussed later.

Reporting

The relevant information is extracted from the LP output, sorted, and printed in terms of areas transferred, cut, and replanted. The optimal values for any user's variables are also included.

A summary of the solution can be produced. The cuts, residuals and thinnings for each period are shown in terms of areas, volumes, average and oldest ages, and the corresponding product/resource or thinning/maintenance totals. For each period, the areas of new planting, the areas replanted into the same crop type, replanted into a different crop type, and those left unplanted, are also given.

An interactive report generator can be used for more detailed output. This is a modification of IFS in which cutting in each period is performed automatically according to the LP solution. All the control and report commands of IFS are available.

Implementation and computational experience

To facilitate experimentation and upgrading of the system, FOLPI has been implemented as a sequence of programs that communicate through intermediate files. Most of the programs of the initial version were written in Basic to reduce the development time, with a few in Fortran and Pascal. The execution time of these is small compared with the time used by the LP package. FOLPI runs on an ICL 2980 computer. Conversion to a VAX 780 in the new N. Z. Forest Service network is underway.

A first program takes the problem formulation (described above) and the input data files, does the lexical and syntactical analysis, expands ranges, and outputs an intermediate code.

The second program generates the LP formulation using the intermediate code and the data files, and produces the LP input in a format acceptable to the LP package. This is the most complex program in FOLPI. Linked list structures are used to build up the LP problem, which must be given by columns to the LP package. Also, if GUBs are used, all the variables in each GUB must be contiguous.

The LP is solved using either LP2900 (ICL 1980) or MINOS (Murtagh and Saunders 1977, 1978). The relevant solution information is extracted from the LP output by another program, printed out, and stored in a file that is accessed by the reporting programs. One reporting program produces a summary, while a second one, obtained by modifying IFS, can be used interactively to display or print more detailed information.

In addition, there is a program that computes optimal unconstrained solutions, using straight-forward stand-level search techniques. This is useful for obtaining a reference objective function value. It can also determine the discounted present value by crop type and age class at the end of the planning period, assuming that no user constraints are effective afterwards. The appropriate objective function terms are automatically generated, and can be merged into the problem formulation file to reduce boundary effects. It is also possible to produce an advanced LP basis corresponding to the unconstrained solution. An output file compatible with FOLPI's report programs is produced.

Some computational results are shown in Table 1. All CPU times are for an ICL 2980 computer, and exclude program loading time. MINOS runs approximately 2.5 times slower on a VAX 780.

As may be expected, computing time rises sharply with the problem size. Forest managers in New Zealand have been running very detailed IFS simulations, with one-year age classes and time periods, and many crop types. FOLPI runs at this level of detail would be expensive, so that some aggregation, at least using two-year age classes, would be necessary with the current version. On the other hand, FOLPI's performance would appear to be competitive with LP-based systems available elsewhere.

The poor performance of the GUB formulation was surprising, since the exploitation of special structures in the LP matrix is supposed to reduce considerably the computational effort.

It is known that with the LP2900 System "... there is a problem of slow convergence in PRIMAL or DUAL for models which are highly degenerate. The objective and sum of infeasibilities only very occasionally move by a non-zero amount sometimes resulting in an excessive number of iterations to solution" (ICL 1981). This is the case here; there are long runs of iterations where the value of the objective function does not change. The same situation occurs with MINOS. Other LP packages will be investigated to see

Table 1: Performance Statistics

Problem number	1	2	3	4 ¹
Periods	8	30	40	40
Age classes	7	25	60	60
Crop types	3	7	11	11
User constraints	13	42	1	39
LP rows (incl. GUBs)	83	638	1542	1580
LP columns	242	1795	11460	11468
GUBs	45	385	1100	1092
LP matrix density (no GUBs)	5.3%	0.64%		0.29%
LP2900 with GUBs:				
Iterations to feasibility	62	217	239(230) ²	>1969
Total iterations	207	564	1714(1383) ²	
Major iterations	91	235	677(540) ²	
CPU time, seconds	24	269	3016(2450) ²	>3600
LP2900, no GUBs (direct/GUBs removed) ³ :				
Iterations to feasibility	44/—	156/170		—/649
Total iterations	162/—	520/596		—/>3143
Major iterations	72/—	213/245		
CPU time, seconds	16/—	159/176		—/>3600
MINOS (no GUBs) (direct/GUBs removed) ³ :				
Iterations to feasibility	64/83	1056/—		—/4702
Total iterations	162/172	1742/—		—/>6464
CPU time, seconds	12/—	347/—		—/>3600

Notes:

1. Runs for problem 4 were abandoned after a time limit of 60 min CPU time.
2. Values in parenthesis are for columns reordered oldest age classes first.
3. First values are for LP generated directly without GUBs, second values are for GUBs converted to ordinary rows with LP2900's DEGUB option. The two LP problems differ only in the ordering of the columns.

if faster convergence can be attained. The high degree of degeneracy also suggests that an appropriate ordering of the columns in the LP matrix could reduce processing time, and some indications of this appear in Table 1.

Limited experience with using the unconstrained solution for the initial basis suggest that there is little or no improvement in the solution time. Further tests are needed to confirm this.

Future work

It is obvious that there is ample room for improvements on the current version of FOLPI.

It is intended to add a number of extensions to the problem formulation “language”. A great deal of repetition of slightly different text strings is involved in setting up the objective function and constraints. It is not difficult to generate these by using a good text editor, but a simpler alternative is desirable. Facilities for easing the problem specification, through summing and looping constructs (Fourer 1983) and/or a macro processor will be added. More extensive error checking and diagnostics will be incorporated into the translator. Solution reporting may also be improved, and appropriately interpreted information obtained from the dual prices and reduced costs could be included. If research currently underway on modelling the transport and utilisation aspects results in a generally accepted LP formulation, features implementing this could be incorporated into the specification language. Nonlinear and Integer Programming capabilities may also be added.

Some unsatisfactory features, such as the different treatment of “yield” tables and other product/resource tables, have been included in FOLPI in order to maintain compatibility with IFS. Ultimately, new cleaner versions of IFS and FOLPI will be implemented. An option for increasing the age class width and period length in later periods would be useful.

On the LP solution side, a number of alternatives should be tried. LP packages other than LP2900 and MINOS will be tested to see if they can better handle the degeneracy problem. Several specialised solution techniques, taking advantage of the special structure of the problem, are worth trying.

It is clear that the bulk of the constraints in FOLPI deal with a network flow, with a relatively small number of general side constraints (the user-supplied constraints), and possibly some user-supplied side variables. This is precisely the kind of problems that algorithms such as those in Glover and Klingman (1981) and McBride and Gupta (1982) are designed to solve. Large computational advantages over general-purpose LP packages have been claimed. At present, however, the implementations are either proprietary (Glover, pers. com.) or not yet ready for general use (McBride, pers. com.). If any of these programs become available it is intended to try

them on FOLPI, although any optimism should be perhaps tempered by the experience with Generalised Upper Bounding.

Another “obvious” approach would be LP decomposition. A number of proposals using the Dantzig-Wolfe decomposition principle have periodically appeared in the forestry literature, and some of them have been implemented on an experimental basis. However, practical experience with decomposition has not been encouraging (Orchard-Hays 1973, Ho and Loute 1981, 1983). It is unlikely that decomposition can be more efficient than large-scale simplex codes, unless the structure of the subproblems is such that they can be solved efficiently with specialised algorithms. FOLPI can be solved using a column-generation approach, based on Dantzig-Wolfe decomposition, where the subproblem can be reduced to finding minimal cost paths in an acyclic network. The fact that this subproblem is very easy to solve (e.g. Wagner 1969, p.235), could result in an efficient computational procedure. Work is in progress on an implementation of this method, through a modification of MINOS.

The unconstrained optimisation gives a solution that is dual-optimal. A variation of the primal-dual algorithm could take advantage of this, possibly combined with one of the approaches discussed above.

Conclusions

FOLPI would appear to be easy to understand and to use, and the possibility of combining optimisation and simulation should contribute to better decision-making. The current version may be viewed as a prototype, and a number of enhancements can be expected.

The computational efficiency of FOLPI, even in its current form, seems satisfactory in comparison with other approaches. There seems to be good potential for improved performance through the use of specialised algorithms.

Acknowledgments

I wish to thank B. Manley, A. Gordon, J. Threadgill, and W. Deadman for helpful suggestions, system testing, and/or programming assistance. Part

of this work was done in 1972–73 while the author was with the Instituto Forestal, Chile.

References

- BOURGAU, J. M. and POUPARDIN, D. 1972. Modèles de gestion sylvicole. *Rev. For. Française* 24, 131–140.
- FOURER, R. 1983. Modeling Languages Versus Matrix Generators for Linear Programming. *ACM Trans. Math. Software* 9, 143–183.
- GARCIA, O. 1981. IFS, An Interactive Forest Simulator for Long Range Planning. *N. Z. Journal of Forestry Science* 11, 8–22.
- GLOVER, F. and KLINGMAN, D. 1981. The Simplex SON Algorithm for LP/Embedded Network Problems. *Math. Prog. Study* 15, 148–176.
- HO, J. K. and LOUTE, E. 1981. An Advanced Implementation of the Dantzig-Wolfe Decomposition Algorithm for Linear Programming. *Math. Prog.* 20, 303–326.
- HO, J. K. and LOUTE, E. 1983. Computational Experience with Advanced Implementation of Decomposition Algorithms for Linear Programming. *Math. Prog.* 27, 283–290.
- ICL. 1980. LP2900 Applications Manual. International Computers Ltd., Technical Publication 6955.
- ICL. 1981. Program Product Set Notice. 2900 Mathematical Programming System (50), 21 Apr 1981. International Computers Ltd.
- JOHNSON, K. N. and SCHEURMAN, H. L. 1977. Techniques for Prescribing Optimal Timber Harvest and Investment under Different Objectives — Discussion and Synthesis. Forest Science Monograph 18.
- MURTAGH, B. A. and SAUNDERS, M. A. 1977. MINOS User’s Guide. Stanford University, Systems Optimization Laboratory, Tech. Rep. 77–9, Revised January 1981.
- MURTAGH, B. A. and SAUNDERS, M. A. 1978. Large Scale Linearly Constrained Optimization. *Math. Prog.* 14, 41–72.

- McBRIDE, R. D., and GUPTA, A. 1982. Solving Generalized Network Problems. Presented at the Detroit TIMS/ORSA Meeting, April 18–21, 1982.
- ORCHARD-HAYS, W. 1973. Practical Problems in LP Decomposition and a Standardized Phase I Decomposition as a Tool for Solving Large Scale Problems. *In*: Himmelblau, D. M. (Ed.) “Decomposition of Large-Scale Problems”, North-Holland, Amsterdam, p.153–166.
- WAGNER, W. 1969. Principles of Operations Research with Applications to Managerial Decisions. Prentice-Hall, Englewood Cliffs, N.J.

September 1984