Automating the Generation of Forest Management Prescriptions

Arif Ur Rahman, Susana Miguel Barreiro, Jos´e Borges
Centro de Estudos Florestais
Instituto Superior de Agronomia
Universidade de Lisboa
Portugal
badwanpk@fe.up.pt, smb@isa.ulisboa.pt, joseborges@isa.ulisboa.pt

ABSTRACT: *Forest management involves treatment of forest stands to improve the current stand structure and condition to achieve management goals. A treatment may involve different operations including plantation, thinning and harvest. StandSim is used to simulate the growth of forest stands for a given time horizon. It requires a file with all the prescriptions for a given situation. A prescription is a set of k cycles, each one consisting of a different permutation of stand variables.*

This paper reports the development of a simple to use and platform-independent tool i.e. Prescriptions-Generator, for automating the generation of prescriptions for forest management. The tool is developed in Java and can be embedded in other software. The embedding in other software is made easy by the use of an XML file for giving input to the tool. It can also be used independent of other tools using a simple to use graphical user interface.

1. Introduction

A forest stand is the basic unit of forest mapping. It contains a group of trees that are more or less homogeneous with regard to species composition, density, size, and sometimes habitat. The goals of forest management are often conflicting. They may include biodiversity maintenance, forest sustainability, maintaining required harvest levels, land conversion and forest protection from fire and diseases. A change in the goals of forest management require a change in the prescriptions applied to forest stands. A prescription is a treatment of a forest stand to change the current stand structure and condition for achieving management goals [1]. A prescription may involve treatments like thinning, shoots selection and harvesting. All trees in a stand may have the same age (even aged) or they have have different ages (un-even aged). Forest managers may want to change un-even aged stands to even aged. They may also want to give some time to uneven age stands for the trees to grow before a clear cut and new plantation. Moreover, they may want to change the specie based on market demands. Different species require different prescriptions. For example Eucalyptus may re-grow once harvested but other trees like pine does not re-grow. The difference in ages of species may also be a constraint for applying different prescriptions. Moreover, there may be different forest management approaches (FMA) based on even aged, un-even aged, stands and new plantations [2].

Eucalyptus is an evergreen and fast growing plant. Among many other industries which use Eucalyptus as a raw product are the pulp, paper and cardboard industries. Eucalyptus is managed in rotations which means that it is planted once and typically it regrows three to five times [3]. Each rotation may involve the implementation of different prescriptions. A prescription may affect one or more stand variables such as stand density, tree age, shoots selection, stand structure and specie composition.

This paper presents a tool for automatic generation of all possible prescriptions considering different variables for managing Eucalyptus stands. The tool is developed in Java and is used along with other tools to identify the set of best prescriptions for maximizing profits and minimizing costs. The other tools include the StandSim simulator and Optimizer.

2. The Standsim Simulator And Optimizer

StandSim is a prescription driven stand level forest simulator that integrates a set of different forest growth and yield models for the most important Portuguese tree species allowing the growth simulation under different user-defined management scenarios [4], [5]. It requires different types of input parameters including forest management data, economic data, information on assortments and stand prescriptions. The prescriptions would be generated manually in Microsoft Excel before the development of the prescription-generator. However, the manual generation of prescriptions was a difficult process and it was difficult to verify if all the required prescriptions were generated. Moreover, the situations in which a large number of prescriptions were generated were impossible to test because of the limitations of Microsoft Excel. StandSim produces results are of two types:

1) detailed characterization of stands growth and yield for the planning horizon;

2) essential information required to run decision models.

The Optimizer is a tool that encapsulates exact mathematical techniques, namely mixed integer programming, linear programming and goal programming. It takes the output of the StandSim as an input and identifies the best prescriptions for a given situation.

3. Prescriptions Generator

The prescriptions are based on many variables including the number of plants planted in the first rotation, the number of shoots left on each stump to grow in the subsequent rotations, forest management alternative (for even and un-even aged stands) and the harvest age. Moreover, the range of the harvest age, shoots and plantation density may increase or decrease the number of prescriptions for a given situation. In addition to this, users may choose from four options. The options are whether subsequent plantations are allowed or not, the stand density, number of shoots and the harvest age are kept constant or they may vary along the planning horizon in each prescription.

The total number of prescriptions for a given situation depends on all the variables, the four constraints and the time horizon. The number of prescriptions for a given time horizon will be the maximum if all the variables are set false and minimum if all the variables are set true. When all the variables are set true, it means that for all cycles in a prescription the harvest age must be the same and there can be no consecutive plantations. Moreover, the plantation density and the number of shoots selection are kept constant in each cycle of a prescription.

3.1 Input and Output
It was decided to make the tool as much compatible with other tools as possible. Therefore, two different moods of input were developed.

3.1.1 XML input
The tool may be embedded in other software and may take direct input from them. This is possible because the prescriptions generator uses a XML input file. A schema was developed against which the input file is validated. Different constraints including data type and range constraints are placed on the values in the XML Schema Definition (XSD) file. The structure of the XML file is as follows:

The structure of the XML is kept simple to the most possible level. Moreover, in the future the addition of other parameters required by other species will be possible. This will help in adding new functionality to the tool without the need to change the existing structure of the code.

```
┌─────────────────────prescriptions.xml─────────────────────┐
│ <?xml version = "1.0" encoding = "UTF-8"?>                 │
│ <prescriptions>                                            │
│ <specieCode>EC</specieCode>                                │
│ <specieName>Eucalypt</specieName>                          │
│ <horizon>100</horizon>                                     │
│ <numOfRotations>3</numOfRotations>                         │
│ <stoolThinningYear>2</stoolThinningYear>                   │
│ ...                                                        │
│ ...                                                        │
│ <fmasToProcess>                                            │
│ <firstFMA>3</firstFMA>                                     │
│ <secondFMA>4</secondFMA>                                   │
│ <thirdFMA>8</thirdFMA>                                     │
│ </fmasToProcess>                                           │
│ <minHarvestAge>9</minHarvestAge>                           │
│ <maxHarvestAge>10</maxHarvestAge>                          │
│ <stepHarvestAge>1</stepHarvestAge>                         │
│ <minNumShotsPerSt>1.2</minNumShotsPerSt>                   │
│ <maxNumShotsPerSt>1.2</maxNumShotsPerSt>                   │
│ <stepNumShotsPerSt>0.2</stepNumShotsPerSt>                 │
│ <minStandDensity>1400</minStandDensity>                    │
│ <maxStandDensity>1400</maxStandDensity>                    │
│ <stepStandDensity>50</stepStandDensity>                    │
│ <sameStandDensity>false</sameStandDensity>                 │
│ <sameNumShotsPerSt>false</sameNumShotsPerSt>               │
│ <noConsecutivePlts>false</noConsecutivePlts>               │
│ <sameHarvestAge>false</sameHarvestAge>                     │
│ <minNyFMA3>10</minNyFMA3>                                  │
│ <maxNyFMA3>20</maxNyFMA3>                                  │
│ <stepNyFMA3>5</stepNyFMA3>                                 │
│ <minNyFMA8>10</minNyFMA8>                                  │
│ <maxNyFMA8>20</maxNyFMA8>                                  │
│ <stepNyFMA8>5</stepNyFMA8>                                 │
│ ...                                                        │
│ ...                                                        │
│ </prescriptions>                                           │
└────────────────────────────────────────────────────────────┘
```

The <fmasToProcess> contains three sub elements which allows a user to choose which FMAs should be added in the prescriptions file. There are four valid values for now i.e. 0, 3, 4 and 8, which can be entered in the three FMA elements. FMA 3 means prescriptions for uneven aged stands, FMA 4 means prescriptions for even aged stands and FMA 8 means prescriptions

for new plantations. The value 0 is used in case it is required to generate prescriptions for less than three FMAs. Therefore, the value of each sub-element of <fmasToProcess> can either be 0 or one of the rest of the values. However, each of the 3, 4 and 8 cannot appear more than once. Moreover, some other constraints are placed like the minimum harvest age cannot be greater than the maximum harvest age for validating user input.

3.1.2 Graphical User Interface: The graphical user interface (GUI) is developed to help use the tool independent of other software. The GUI is developed independent of the XML schema and the XML file. It provides a simple to use form for getting input from users. In case the GUI is used then the XML and the XSD files are not required.

Users may choose from two different output moods. The output moods are as follows.

1) CSV File: The tool can generate a comma separated (CSV) file. This option can be helpful in cases where number of prescriptions are small. The number of prescription is small when all the four constraints are set to true. Moreover, some already developed tools like the simulator presented in Section II requires a CSV file. The generation of a file is also helpful in case it is required to keep the file for future reference. However, the generation of a CSV file may not be a good option in case the number of prescriptions is large i.e. millions.

2) Processing without Storing: There may be situations in which all the prescriptions for a given time horizon are generated to find the best ones. The number of prescriptions may be in millions and therefore storing them in a file may not be an option. In this case a prescription is generated and then sent to the other tool (simulator) for processing. The process of repeats till all the prescriptions are processed. In this option users may need to configure the source code of their software to integrate the prescription-generator.

3.2 Generating the Prescriptions
A prescription is composed of k cycles, each one corresponding to an FMA. For instance, for a eucalyptus stand that is in the 4th year of the second cutting cycle, a possible prescription for a planning horizon of 50 years may be made of 6 cycles: cycle 1 - apply FMA (4,1) and harvest at 10 years of age; cycle 2 - regenerate the stand by coppice, apply FMA (4,2) and harvest at age 12; cycle 3 - regenerate the stand planting 1250 trees per hectare, applying FMA (4,1) and harvesting at age 10, etc. However, forest management is not completely defined until further details are provided for some of the matrix operations, namely plantation, thinning and final harvest. More information required include the minimum and maximum values of the number of trees at plantation per hectare, the age for harvesting, number of cutting cycles and the periodicity and intensity of thinning are also required. The prescriptions result from building sequences of matrices representing all possible permutations generated based on the user-defined intervals. A stand can be managed according to one or many of the generated prescriptions.

The total number of prescriptions for a given situation depends on the time horizon, the constraints on keeping the same harvest age, the number of rotations, same number of plantation in the first rotation and same number of shoots selection in the next rotations in each cycle. The time horizon plays an important role in the number of prescriptions. The longer the time horizon, the more cycles in each prescription and thus a greater number of prescriptions. The maximum number of prescriptions are generated in case all the constraints are false and minimum number of prescriptions are generated in case all the constraints are true.

Users may choose combining different FMAs e.g. for the first thirty years a stand is managed as un-even aged but then it is converted to even aged. Another situation may be that for the first twenty years there is nothing in a stand but then it is planted. If multiple FMAs are used then it is verified that the same number of prescriptions is generated for each FMA.

4. Testing

The prescriptions-generator was tested in two different ways. First, the prescriptions were tested by group members along the development period. Second, the tool was used in practical situations and prescriptions were generated to be used by the StandSim simulator. Different situations were tested like the smallest number of prescriptions were generated and compared with prescriptions generated for the same criteria by a different group member. The tool was also tested for complex situations in which millions of prescriptions were generated.

In the testing process the largest file generated with the tool was approximately 84 Gigabytes and it took around two hours to complete the procedure on a latest computer. Of course the file could not be opened by any other tool but it proved that the tool

did not hang or break in generating even such a large number of prescriptions.

5. Conclusion And Future Work

Generating prescriptions for different species depend on different variables. Though, some of the variables may be the same for different species but they may have a different range of values. The number of prescriptions depend on the number of variables involved in each prescription and the range of values they can have. There may be situations for which the analysis of millions of prescriptions may be required. For such situations the use of sophisticated decision support systems is required which help in choosing a few feasible prescriptions for stand management.

In the future the functionality of converting stands from one specie to another along a time horizon will be added. This will require the addition of more elements in the XML and changes in the GUI.

References

[1] P. Bettinger, K., Boston, Siry, J. P., Grebner, D. L. (2010). Forest management and planning. Academic Press.

[2] Duncker, P. S., Barreiro, S. M., Hengeveld, G. M., Lind, T., Mason, W. L., Ambrozy, S., Spiecker, H. (2012). Classification of forest management approaches: a new conceptual framework and its applicability to european forestry, Ecology and Society, 17(4).

[3] Larranaga, A., Basaguren, A., Pozo, J. (2009). Impacts of eucalyptus globulus plantations on physiology and population densities of invertebrates inhabiting iberian atlantic streams, *International Review of Hydrobiology*, 94 (4), p. 497511, August.

[4] Faias, S. P., Palma, J. H. N., Barreiro, S., Paulo, J. A., Tom´e, M. (2012). Resource communication. sIMfLOR platform for portuguese forest simulators, *Forest Systems*, 21 (3), p. 543–548.

[5] Faias, S., Palma, J., Barreiro, S., Paulo, J., Tom´e, M. (2012). sIMfLOR Portuguese forest simulators platform, Instituto Superior de Agronomia, Bordeaux, France, Tech. Rep.