Improved procedures for maritime pine management with ModisPinaster using the ModisOptimizer tool

Gilles Le Moguédec[1], Teresa Fidalgo Fonseca[2], Francois de Coligny[1], Adelaide Cerveira[2,3]

[1]UMR AMAP, France
[2]University of Trás-os Montes e Alto Douro, Portugal
[3]INESC TEC, Portugal

Introduction

A brief presentation of ModisPinaster

Growth model with diameter distribution for Pinus pinaster

- **Inputs from current inventories (stand level variables)**
- **Several outputs (e.g. volume, biomass per compartment, carbon, energy)**

Flexible description of the diameter classes (Johnson S_B distribution)

Inclusion of a mortality module (competition and wind damages)

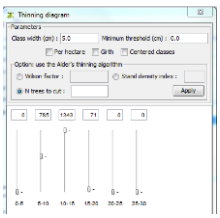
Thinning simulation modules

Interactive or automatic procedure, based on number of trees, or density indices (Wilson factor, Fw, or Reineke, SDI). Automatic management also available for fixed ranges of Fw or SDI.

The whole model (including mortality module) is deterministic.

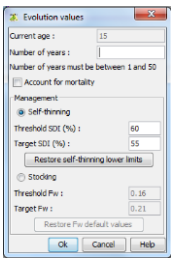
Using ModisPinaster

Manual thinning



and

Evolution between 2 dates



Stand Density Index

$SDI\% = N(25/dg)^{1.897}/SDI_{max}x100$

Wilson Factor

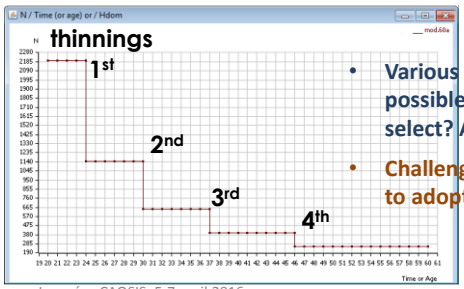
$Fw = 100/(N^{0.5}hd)$

N: trees.ha^{-1}

h_d: dominant height (m)

d_g: quadratic mean diameter (cm)

One possible simulated trajectory:

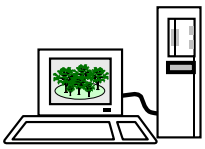


- **Various options → Good! But with a large set of possible simulated trajectories which option to select? Arduous task to perform manually…**
- **Challenge → To use it more efficiently, it was logical to adopt an optimization approach**

ModisOptimizer: an optimisation tool for Modis Pinaster

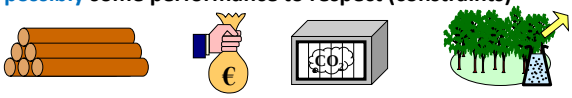
Elements needed for Optimization

- **A growth model:**



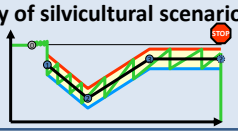
Here : **ModisPinaster**

- **An evaluation criterion to optimize (objective function) and possibly some performance to respect (constraints)**

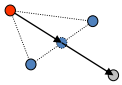


Here :
- **merchantable volume**
- **aboveground biomass**

- **A family of silvicultural scenarios (control variables)**



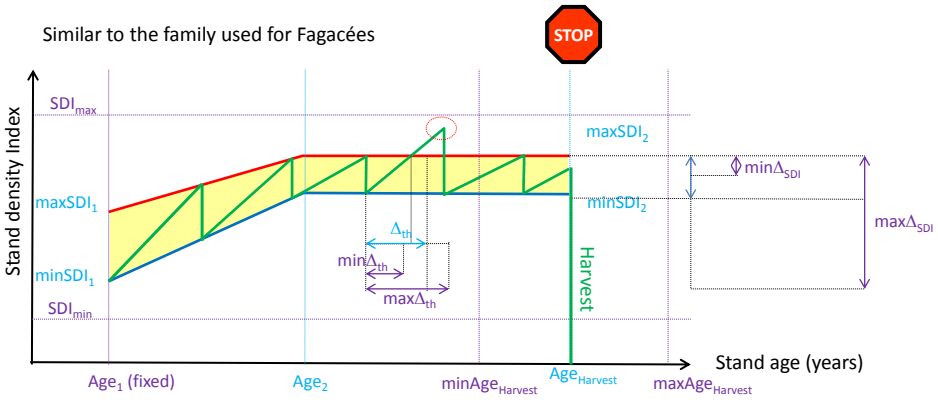
- **An adapted optimisation method**



Here : **Nelder-Mead algorithm** (Cf. Caqsis 2015)

The family of silvicultural scenarios

Similar to the family used for Fagacées



A scenario is defined by 7 control variables, all bounded, some of them ordered (Cf. Caqsis 2015).

But for SDI we have double constraints : that need special treatment.

$$\begin{cases} SDI_{min} \le SDI_1 \le SDI_2 \le SDI_{max} \\ min\Delta_{SDI} \le SDI_2 - SDI_1 \le max\Delta_{SDI} \end{cases}$$

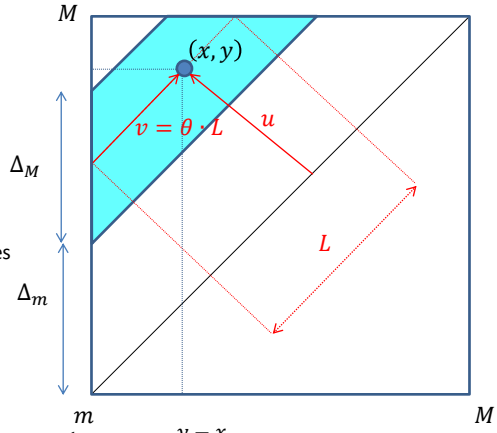
Using double-constrained coordinates

Problem:
how to generate a point (x, y) so that :

$$\begin{cases} -\infty < m \le x \le y \le M < +\infty \\ 0 \le \Delta_m \le y - x \le \Delta_M < +\infty \end{cases}$$

Where $\begin{cases} -\infty < m < M < +\infty \\ 0 \le \Delta_m < \Delta_M \\ \Delta_m + \Delta_M < M - m \end{cases}$

Response:
By using a bijective transformation of coordinates so that the new coordinates are each bounded and do not appear in the same constraint.



A solution is:

$$\begin{cases} x = m + \theta\left[(M-m) - \sqrt{2}u\right] \\ y = m + \theta(M-m) + (1-\theta)\sqrt{2}u \end{cases}$$

where $\begin{cases} \frac{\Delta_m}{\sqrt{2}} \le u \le \frac{\Delta_m + \Delta_M}{\sqrt{2}} \\ 0 \le \theta \le 1 \end{cases}$

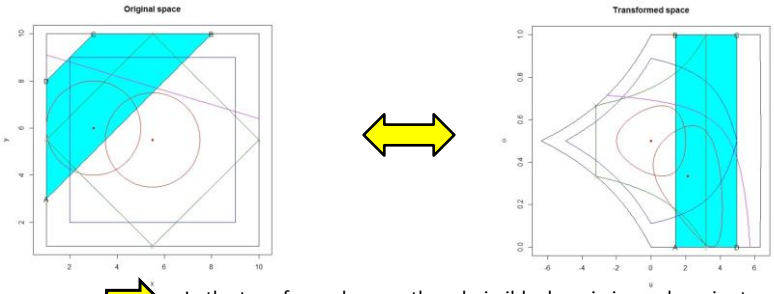
$\Leftrightarrow$

$$\begin{cases} u = \frac{y-x}{\sqrt{2}} \\ \theta = \frac{x-m}{(M-m)-(y-x)} \end{cases}$$

where $\begin{cases} m \le x \le M - \Delta_m \\ m + \Delta_m \le y \le M \\ \Delta_m \le y - x \le \Delta_M \end{cases}$

Illustration of transformation

Representation of the admissible domain and some geometrical shapes:



In the transformed space, the admissible domain is much easier to manage

This approach can be extended to the n-dimensional case:

$$\begin{cases} -\infty < m \leq x_1 \leq x_2 \leq \cdots \leq x_n \leq M < +\infty \\ \forall i \in \{1, \cdots, n-1\}: 0 \leq \Delta_m \leq x_{i+1} - x_i \leq \Delta_M < +\infty \end{cases}$$

Where the parameters are so that all constraints are necessary to delimit the admissible domain.
(If it is not the case, another transformation is used)

Optimisation

Thanks to previous transformation, it is easy to generate a set of control variables
- that is always in the admissible domain ;
- and allows to explore the whole domain.

Optimization is performed with the Nelder-Mead algorithm that works on unbounded variables.

Another bijective transformation is used to convert bounded variables in unbounded ones that are used by the algorithm and, once the Nelder-Mead solution found, to perfom the reciprocal transformation in order to obtain interpretable variables (Cf. Caqsis 2015).

Generic tools for that have been implemented in the capsis.lib.optimisation library.

This library contains also the skeletton of the function to optimize.

To build ModisOptimizer, only a few java classes specific to ModisPinaster have been written. They allow to easily define the problem to solve (objective function, definition of admissible domain, writing the report….

Due to the fact that there is no randomness in ModisPinaster, optimization with ModisPinaster run a lot faster than for Fagacées. Only 1-2 minutes in general to obtain a solution.

That allows to study a whole set of new questions with ModisPinaster

Applications for the Optimization tool

Example

Input data set: t =15 yrs, hd = 7.8 m, N = 2200 trees.ha^{-1}, G = 12.9 m^{2}ha^{-1}, values for a maritime pine stand of average site index, according to the Stand Tables (North of Portugal).

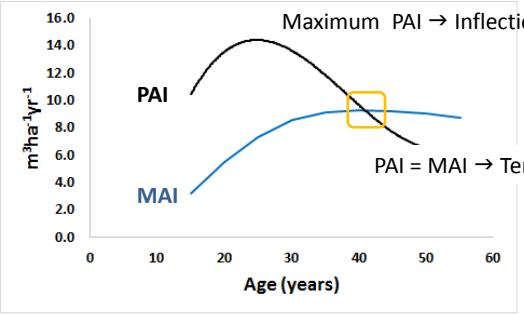
Traditional silviculture model (Moreira and Fonseca, 2002)

Before thinning						Thinning				After thinning				Total		
t	hd	N	dg	G	V	N	dg	G	V	N	dg	G	V	VT	MAI	PAI
15	7.8	2200	8.6	12.9	47.9									47.9	3.2	
20	10.3	2200	11.5	22.9	109.8	433	9.7	3.2	10.4	1767	11.9	19.7	99.4	109.8	5.5	13.4
25	12.5	1767	14.7	29.9	171.9	554	12.6	6.9	37.3	1213	15.6	23.1	134.6	182.2	7.3	14.6
30	14.4	1213	18.3	31.7	208.0	295	15.0	5.2	32.9	918	19.2	26.5	175.1	255.7	8.5	13.6
35	16.0	918	21.3	32.7	238.2	179	17.0	4.1	27.8	738	22.2	28.7	210.4	318.7	9.1	11.6
40	17.5	738	23.9	33.2	262.9	119	18.8	3.3	23.8	619	24.8	29.9	239.1	371.2	9.3	9.6
45	18.8	619	26.1	33.2	282.6	84	20.6	2.8	20.8	535	26.9	30.4	261.8	414.7	9.2	7.9
50	20.0	535	28.0	32.9	297.4	62	22.2	2.4	18.2	472	28.7	30.5	279.2	450.3	9.0	6.4
55	21.1	472	29.5	32.4	307.7	48	23.6	2.1	16.0	425	30.1	30.3	291.7	478.9	8.7	

- A **thinning** each **5 years** of period
- **Rotation** lenght equal to **55 years** , corresponding to a total volume **VT=478.9** m^{3}ha^{-1}
- **Thinning** weight defined by Fw = 0.23 (SDI values in the mean range of 30% - 35%)

The stakeholders (pine industry) point for the need of a new silvicultural model !

Classical elements for an optimal silvicultural schedule



- **Traditional rotation lenght: associated to the term of explorability**
  35 – 50 yrs (depending on site quality)
- **Increasing demands of the industry point to shorter rotation lenghts in the future**
  15 – 20 yrs (before attaining decreasing values of PAI**)**

New questions that can be answered with the Optimizer

- **Would the traditional silvicultural model, that has been prescribed, corresponds to the "optimal" solution?**
- **Does it provides the maximum achievable timber?**
- **Is there any major differences when managing for different aims (use of different objective function)?**
- **How many thinnings and when to thin, if any, should be performed to maximize a selected objective function (volume, biomass, energy…)?**
- **How does the best solution found by the optimizer change when varying the rotation length?**

Example – Models of silviculture (scenarios) tested

Objective function: *Merchantable volume* **(V)**

Objective function: *Aboveground biomass* **(B)**

Maximum rotation age:

- 50 years (long) → **Scenario V_l**, **Scenario B_l**
- 20 years, (short) → **Scenario V_s**, **Scenario B_s**

Total of 4 scenarios

Main results obtained with the Optimizer

- The traditional silvicultural model does not correspond to the "optimal" solution. Maximum values of timber would be achieved by keeping the stand in higher densities in comparison to the traditional model where the total volume at 50 yrs, reported in the Stand Tables, is VT= 450 m^3ha^{-1} with 6 thinnings.
- No major differences in total yield or either in silvicultural activities were found when managing for the different objective function: merchantable volume (**V**) vs aboveground biomass (**B**).

	Total volume (m^3ha^{-1})	Total Biomass ($t\ ha^{-1}$)
Scenario V_l	**721**	**294**
Scenario B_l	**718**	**295**

Just two thinnings: the 1st at 37-38 and the 2nd to age 45. Final cut at 50 yrs.

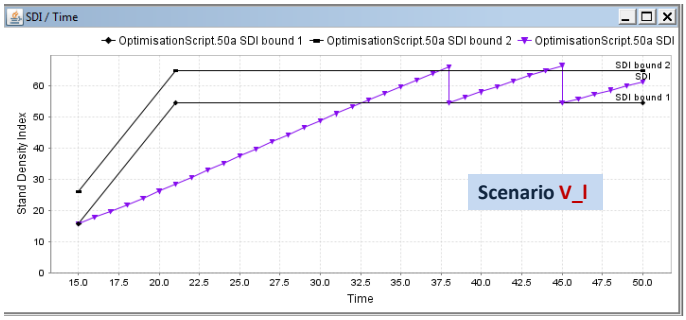
Scenario V_s	**107**	**57**
Scenario B_s		

No thinning is performed. Density is lower than the SDI lower limit. Final cut at 20 yrs.

- The Optimizer tool provided the "best" solutions, in a quite short execution time (range of 3s to 1min3s), for the selected management problems.

Main results obtained with the Optimizer

- The Optimizer tool has proved to greatly enhance the ability of ModisPinaster.
- Also it offers the possibility of considering different shapes (width) of the SDI envelop with time to take into account for differences in management during the evolution of the stand (e.g., young stage *versus* maturity). This was noticed although did not influenced the results as the initial stand used as a case study had a low density value, hence, no thinning needed to be performed in the earlier stage of development. It should be of importance for future tests.



Conclusion

What's new with ModisOptimizer?

With the new tool added to ModisPinaster, ModisOptimizer:

- **The user can decide and change the criterion to maximize (e.g. total or merchantable volume; biomass per compartment or total; carbon; energy)**
- **The program determines accordingly the silvicultural operations to perform:**
  - **Age of thinning;**
  - **Number of thinning;**
  - **Intensity of thinnings;**
  - **Age of clear cut.**

- **The results provided by the ModisOptimizer showed that the tool is of great interest to support the management of maritime pine.**
- **Further tests are under course.**
- **It is advisable to analyze the results.**

Thank you for attention!