Author(s): Eyvindson, Kyle; Rasinmäki, Jussi; Kangas, Annika

Title: Evaluating a hierarchical approach to landscape level harvest scheduling

Year: 2018

Version:

Supplementary material to:

Evaluating a hierarchical approach to landscape level harvest scheduling by Kyle Eyvindson, Juusi Rasinmäki and Annika Kangas

This supplementary document uses synthetic data to test to computational requirements and performance of the different hierarchical models. The methods and data used in this analysis are held in a GitHub repository at https://github.com/eyvindson/hierarchical

The entire process was modelled in Python, using the Pyomo package (Hart et al. 2012) to model the problem and link the model to a commercial optimization package (CPLEX 12.6 in this case). A set of 1,000 artificial stands were generated to represent potential initial conditions, and these stands were simulated for 10 years through 6 periods (four one-year periods and two three-year periods). This dataset can be combined to represent a wide variety of forest sizes.

The Python code is partitioned in to six sections:

1. List of variables used throughout code:

    All variables which influence the optimization process and the re-structuring of the data are included here. A change in these variables will increase/decrease the computational complexity of the problem, and increase/decrease the amount of time allocated to the different optimizations.

2. Adding packages

    Here all of the importing of python packages is done.

3. Create optimization functions

    Each of the optimization functions are created with a separate function. Pyomo is used to create the models, and each optimization model reflects the models created in the manuscript.

4. Importing data

    The set of 1000 synthetic stands are imported through Pandas. Users need to ensure that the path and data file name reflects their system.

5. Creating structured data for optimization

    Here the synthetic stands are organized into a data frame which reflect the problem structure. A number of departments are created with a random number of stands (bounded between a minimum and maximum value). The area of each stand is also a random value, bounded between a minimum and maximum value.

6. Solving the optimization problems

Each of the models are sent to the optimization software (in this case CPLEX), and given a set amount of time to process.

To provide context for the target values, the monolithic problem is solve to find the maximum possible amount of timber produced during the entire period, and is resolved using a percentage of that maximum as a target for each period. With a synthetic dataset, setting appropriate targets can be difficult as the specific case changes with the random variables.

Once solved, the objective function is provided, and the time taken to generate the solutions is also provided.

For this supplementary material three cases with increasing complexity were solved, we provide the variables used in table S1 and solutions to each of the problems is found in Figure S1.

Table S1. Adjustment of a selection of variables used in the six different tests.

	Variables adjusted		
	Depts	min_stands	max_stands
Test 1	50	300	500
Test 2	150	300	500
Test 3	300	300	500
Test 4	450	100	200
Test 5	450	200	300
Test 6	450	300	400

The results and performance of the different models are highlighted in Figure S1. The results are similar to that found in the manuscript, with the large problems (when the departments in the analysis are 300) showing that the monolithic problem does not come close to the global solution under the time limits provided. This is seen as the integrated model finds better solutions than what has been produced through the monolithic model.
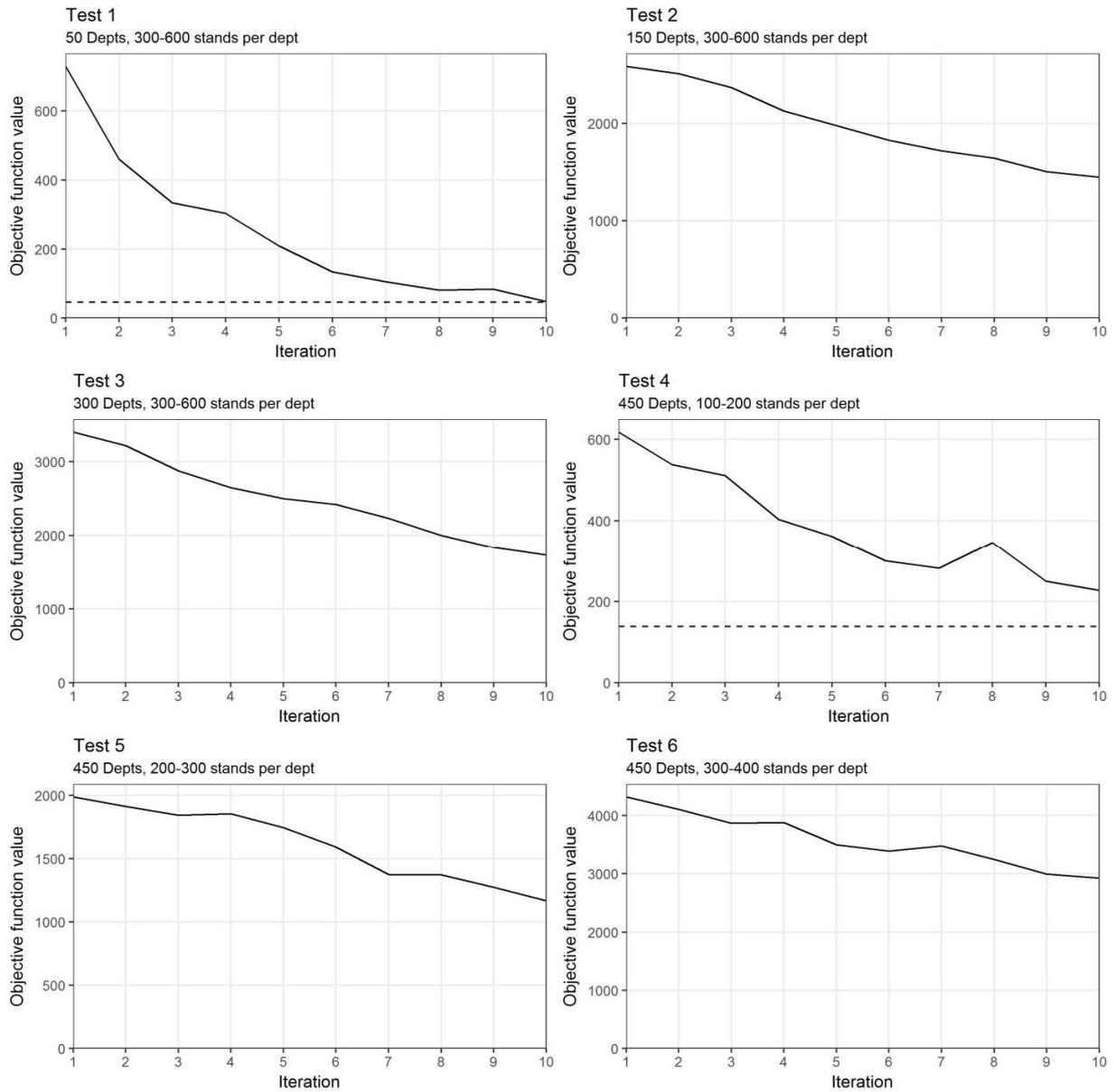
Figure S1. The objective function values for the monolithic solution (dashed line) and the iterative hierarchical solution (solid line) for different iterations. For tests 2, 3, 5 and 6, the monolithic problem did not find a solution within the range of the hierarchical solutions.

References:

Hart WE, Laird C, Watson J-P, Woodruff DL (2012). Pyomo - Optimization Modeling in Python. Springer