Towards Sustainable Timber Harvesting of Homogeneous Stands: Dynamic Programming in Synergy with Forest Growth Simulation

Mario C. López-Locés
Universidad Autónoma de Nuevo León (UANL)
Graduate Program in Systems Engineeringy
mariocesar@lopezloc.es

Roger Z. Ríos-Mercado[1]
Universidad Autónoma de Nuevo León (UANL)
Graduate Program in Systems Engineering
Department of Mechanical and Electrical Engineering
roger.rios@uanl.edu.mx

Oscar A. Aguirre-Calderón
Universidad Autónoma de Nuevo León (UANL)
Department of Forest Science
oscar.aguirrecl@uanl.edu.mx

José Luis González-Velarde
Tecnologico de Monterrey
School of Engineering and Sciences
gonzalez.velarde@tec.mx

January 2020

[1]Corresponding author

Abstract

In this work, the problem of maximizing the volume of wood harvested in a single species stand over a period of time is addressed. To this end, a solution that combines dynamic programming and a single-tree forest growth simulator is developed. In this method, the decision variable of the amount of wood to be harvested at each period is discretized. This ensures that the method finds a global optimal solution within the given discretization. In the past, there have been approaches that use exact methods that solve this problem, but these approaches consider the stand growth as a whole and require the simulator to meet certain conditions. In our work, a single-tree growth simulator is used. With these tools, different alternatives for the parameters of thinning percentage in each period, duration of the planning horizon, and the selection of the trees to be harvested, among others, are explored and assessed. The results showed that the proposed method is useful not only as a tool to optimize the harvesting of the timber of a single-species stand, but to explore different alternatives to the usual practices, that continue to change constantly.

Keywords: Forest management; dynamic programming; forest growth simulation; stand-level optimization; thinning regime.

1 Introduction

One of the decisions in forest management and planning is to determine the frequency and intensity of the treatments applied to a stand during a series of periods, in order to maximize the benefits obtained at the end of a planning horizon, whose duration can span several decades. This problem is known in the literature as the stand-level optimization problem [13]. As the trees in the stand grow continually during the timespan of the planning horizon, it is necessary to take this factor into consideration to deliver a solution that is as close as possible to a real case scenario. Additionally, to prevent the overexploitation of the stand and to promote the regeneration of the forest , the partial harvesting or treatment applied during each period must be subject to regulations that guarantee the preservation of the stand. Tahvonen and Rämö [20] make a comparison between clear-cut and continuous cover regimes, favoring the latter not only in wood extraction, but also in aesthetic and environmental conservation aspects.

The stand-level optimization problem has been a subject of research since the decade of the 1960s, with the work of Chappelle and Nelson [9] as one of the pioneers on this field. Since then, different approaches to solve it have been proposed. Those works have been classified in the literature in three broad categories: Non-linear programming, heuristics and dynamic programming [5, 22].

Dynamic programming is still one of the most popular approaches to solving stand level optimization problems. According to a recent study by Yoshimoto et al. [22] who evaluated different solution methods, they found that the nonlinear program approaches struggle to finding global optimal solution due to the non convexity of the problem. In the case of heuristic methods, they also fail in finding optimal solutions.

Dynamic programming is an optimization method, proposed by Bellman [4], that has found application in different fields, including forest management, with the work of Amidon and Akin [1] as one of the first that use this approach to solve the stand-level optimization problem. In general terms, this method consists of recursively dividing a complex problem into smaller subproblems that, when solved, can contribute to the solution of the other problems in the recursion chain.

Until the decade of the 1980s, it was not practical to use dynamic programming to solve the stand-level optimization problem, given the technological limitations of the time, in terms of computational processing power and storage, and the characteristics of the problem that grows exponentially when the number of decisions and periods increases. This motivated the development of solutions based on dynamic programming that could achieve the optimality of the solution by reducing the dimensionality of the search space, provided that certain conditions of non-concavity of the solution space, dependent of the growth simulator, were met. There are certainly many works based on dynamic programming. For a survey on those methods, the reader is referred to the work by Yoshimoto et al. [22]. Table 1 summarizes some of the most relevant works that use dynamic programming to solve variations of the stand-level optimization problem.

Table 1: Most relevant dynamic programming approaches to the stand-level optimization problem. Column “Reference” cites the author or author and year of the different dynamic programming models proposed. Column “Simulator Characteristics” indicates if the simulator is a single-tree growth simulator or not (whole-stand growth simulator), and if it is distance dependent, that is, if it includes spatial indices for competition [21]. Column “Stand Age” if the age of the trees in the stand is even or not.

References	Simulator Characteristics		Stand Age
	Single tree	Distance dependent	
Amidon and Akin [1]	No	No	Even
Brodie et al. [6]	No	No	Even
Brodie and Kao [7]	No	No	Even
Kao and Brodie [15]	No	No	Even
Chen et al. [10]	No	No	Even
Haight et al. [14]	Yes	No	Even
Paredes V. and Brodie [18]	No	No	Even
Arthaud and Klemperer [2]	No	No	Even
Yoshimoto et al. [23]	Yes	No	Even
Brukas and Brodie [8]	No	No	Even
Bettinger et al. [5]	Yes	No	Even
Diaz-Balteiro and Rodriguez [11]	Yes	No	Even
Graetz et al. [12]	Yes	No	Even
Yoshimoto and Marušák [24]	No	No	Even
Asante et al. [3]	No	No	Even
Ribeiro et al. [19]	Yes	Yes	Even
This work	Yes	Yes	Even

Paredes V. and Brodie [18] use this approach to solve the problem of dimensionality with their Projection Alternative TecHnique (PATH), derived either from a network formulation based on the Dijkstra's shortest path algorithm or from generalized Lagrange multiplier theory. Yoshimoto et al. [25], developed an improvement on PATH, called Multi-Stage PATH (MSPATH). They detected that some scenarios in the original PATH were just looking to the next immediate stage and this was insufficient to guarantee optimality. Thus, their proposed algorithm made all possible look-ahead paths to further stages, reducing some of the efficiency of the algorithm, but gaining more robustness in the finding of the optimal solution. This approach also requires the same conditions as the original PATH.

In this article, the problem of determining the percentage of the basal area of a stand that is harvested during a number of periods is addressed. The aim is to maximize the volume of wood collected at the end of the planning horizon, while following the regulations during all the partial thinnings. For this problem, the use of a dynamic programming (DP) model in synergy with a forest growth simulation software, to maximize the volume of wood harvested from a stand during a planning horizon is proposed.

The stand used for the simulation is not static, as it considers the growth and mortality of the trees as well as the competition among them. It is proposed in this work to embed the forestry simulator software BWIN Pro Forest Growth Simulator (`https://www.nw-fva.de`)[17] during the optimization stage, to maximize the amount of wood harvested at the end of the planning horizon. The module of this free software library is capable of simulating a stand at the level of a single tree, while considering the competition among trees, based on the distance in a realistic way. This level of detail allowed us to test different strategies, such as the selection criteria to choose the trees to be harvested on each period, the number of periods per planning horizon, the duration of the planning horizon itself, and the number of cutting options available on each period. The exploration of these strategies allowed us to determine which configurations maximized the volume of timber harvested or reduced the computational time of the simulation. Although the methodology can be applied to any single stand, for our purposes, we apply it to two case studies with *Pinus cooperi* and *Pinus sylvestris*.

The main contribution of our proposed method resides not only on optimizing the volume of wood harvested at the end of a planning horizon but also as a tool to explore different scenarios that the usual practices. This was possible thanks to the inclusion of a forest growth simulation software module that is able to calculate the growth of a stand at the individual tree level and to easily reconfigure the parameters of the proposed solution based on dynamic programming. A first step was discretizing the cutting options into a finite space. This discretization can be very fine to better represent reality. As it explodes exponentially, only a subset of solutions is used. In Section 4, it is demonstrated that this discretization process does not compromise the optimality of the solution. This approach guarantees to find the optimal solution, regardless of the characteristics of the simulator or the stand, as it explores the entire discretized search space, this is, the set of all feasible solutions for an instance of the problem. This provides an advantage with other methods in the literature that consider only a subset of solutions. Existing methods also require that the growth simulator meet very specific properties that are not always achieved.

The organization of this paper is as follows. Section 2 presents the problem description, the mathematical programming model of the problem, and the DP approach for optimizing the harvesting of the stand given during a given planning horizon. Section 4 describes the conditions of the experiments, the computational experiments performed to evaluate the performance of the solution method and the analysis of the results obtained. Finally, Section 5 presents the conclusions reached during this study and suggests some areas that could be interesting to address in future work.

2 Mathematical Framework

2.1 An Optimization Model

From the previous description of the problem we could infer the parameters that are needed for the construction of a model that optimizes the harvesting of a stand, such as the number of periods, the planning horizon in years, the initial state of the stand, and the minimum requirements for a stand to be considered feasible.

The decision to be taken at the start of each period is to determine the percentage of the basal area of the stand to be thinned, and according to that decision we can determine the new state of the stand, the volume of timber harvested, and the growth of the stand at the beginning of the next period. With this information, it is possible to build a mathematical model to determine the sequence of decisions of cutting percentages that maximizes the volume of timber harvested at the end of the planning horizon.

Parameters:

$N:$ Number of periods.

$i = 1, 2, \ldots, N:$ Time period index.

$x_0:$ A three-dimensional vector representing the initial state of the stand.

$\alpha:$ A three-dimensional vector that contains the minimum required values of the number of trees, dominant height, and basal area.

Decision variables:

$w_i:$ A continuous variable that represents the cutting percentage value at the start of period i. For instance, if a decision of harvesting 15% of the stand is taken at period i, then $w_i = 0.15$.

Dependent variables:

x_i : State of the stand after the thinning in period i. This is a three-dimensional vector whose components describe the stand in terms of (1) the number of trees per hectare, (2) dominant height, and (3) basal area.

G_i : Three-dimensional vector describing the growth of the stand in each of its components (x_i) during period i. Similarly to the case of vector x_i, all the attributes of the stand simulator are used.

$V_i(w_i(x_{i-1} + G_i))$: Volume harvested at the start of period i. Clearly, $V_i(\cdot)$ is a function of the state of the system at the end of period $i-1$, given by x_{i-1}, its growth during the past period G_i, and the cutting value percentage w_i.

Mathematical model:

$$\text{maximize} \quad \sum_{i=1}^{N} V_i(w_i(x_{i-1} + G_i)) \tag{1}$$

subject to

$$x_i = (1 - w_i)x_{i-1} + G_i \quad i \in N \setminus \{1\} \tag{2}$$

$$x_i \geq \alpha \quad i \in N \tag{3}$$

$$0 \leq w_i \leq 1 \quad i \in N \tag{4}$$

The objective (1) of this problem is to maximize the volume of harvested wood at the end of the planning horizon considering the natural growth of the trees in the stand. Constraints (2) establish the relationship between the state of the system at the start of period i as a function of the state in the previous state (x_{i-1}), the stand growth (G_i) and the harvested percentage in period i (w_i). Constraints (3) ensure that a minimum stock, usually stablished by government regulations, is met. Finally, Constraints (4) describe the nature of variables w_i. In the model, the value of G_i is dependent on the value of x_{i-1}, or the state of the stand after the thinning at the start of the period $i-1$.

2.2 Dynamic Programming Formulation

Given the sequential structure of the decision process, a popular solution technique is DP. Several DP modelling frameworks have been proposed in the past, one of them is the DP framework on which this work is based [16]. To this end, rather than working with continuous harvesting decisions (w_i) we discretize this variable and use a finite number of cutting options (stored in set Y). This makes the problem more tractable and, if the discretization is fine enough, the optimal solution is very accurate.

Parameters:

Y : Discrete set of cutting options.

The parameters N, i, x_0 and α, are as defined before in Section 2.1.

Decision variable:

y_i : A discrete variable that represents the cutting option at the start of period i, $y_i \in Y$.

Dependent variables:

The dependent variables x_i and G_i, are as defined in the previous model.

$V_i(y_i)$: Volume harvested in period i under cutting option y_i.

$f_i(x_i)$: Maximum volume of wood harvested from the state x_1 of the first period to the period i when the current state is x_i.

DP recurrence relation:

$$f_i(x_i) = \max_{0 \le y_i \le 1} \{V_i(y_i) + f_{i-1}(x_{i-1})\} \tag{5}$$

where

$$x_i = (1 - y_i)x_{i-1} + G_i \qquad i \in N \setminus \{1\} \tag{6}$$

$$x_i \ge \alpha \qquad i \in N \tag{7}$$

The objective of this problem is to maximize the volume of harvested wood at the end of the planning horizon, that is, the optimal solution is given by $f_N(x_N)$. During each period, the state of the forest must maintain a level equal or superior on the attributes: number of trees, dominant height and basal area, represented by vector α. Note that (6) corresponds to the discretized version of (2) and (7) is the same as (3) and just included here for completeness sake.

Figure 1 shows the representations of vector x and its components: number of trees, basal area and dominant height. Where the red area represents the minimum stock value required by Constraint 7. Figure 1 (a) shows the conditions that a stand requires to be considered a feasible solution, where all three components of the vector x must be higher or equal than the values on vector α. Failing to meet any of the three requirements, renders the stand unfeasible.

In this work, it is assumed that a stand is equal to another, just when the three components of the vector x are the same on both of them. For example, Figure 1 (b) shows two stands that have the same number of trees, but that differ in the basal area and dominant height values, thus, the two stands are considered as different.

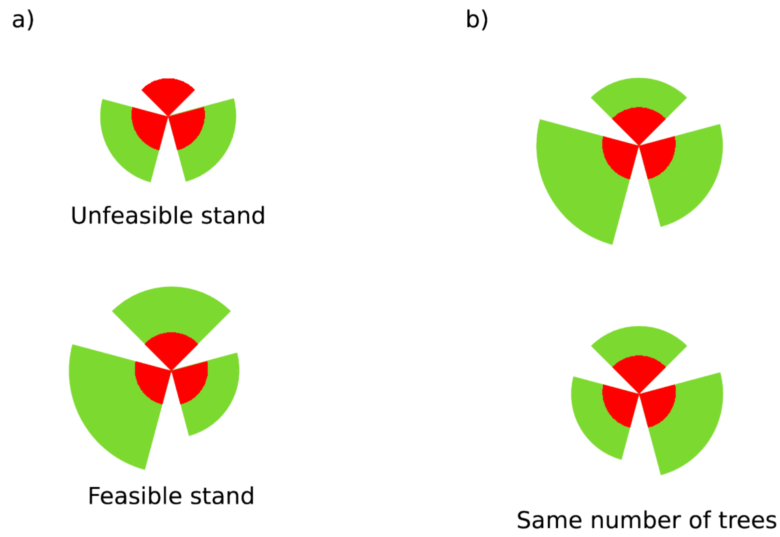

Figure 1: Graphic representation of the DP model. The top wedge represents the number of trees in the stand, the left one its basal area, and the right one, the dominant height of the stand. The green region represents the amount that can be feasibly harvested, and the red area the minumum stock of the stand. If the green area is less than the red area, for any of its wedges, the solution is unfeasible (a). A solution is considered equal to another only if its three values are equal. If any value is different, the solutions are also different, as in the case of (b) where two stands have the same number of trees, but their basal area and dominant height are different.

The graphical representation of the DP model used in this work is shown in Figure 2, where all the variables are depicted here. G_i is the growth of the stand from the state x_{i-1} to the start of the period i, just before the thinning. At $i = 0$, the values of the elements of $G = 0$ are equal to 0 and the initial stand is x_{-1} (recall, from Section 2.1 that G_i and x_i are three-dimensional vectors). The value of G_i is dependent on the value of x_{i-1}, or the state of the stand after the thinning at the start of the period $i - 1$. The variable y_{i+1} is the cutting option selected in this period. The cutting option is the percentage of the basal area of the stand to be harvested and in this model is selected from a finite set of cutting options. For example, if $y_2 = 0.3$, then, during the third period, the basal area of the stand will be thinned by 30%. $V_i(y_i)$ represents the volume of wood harvested from cutting the percentage y_i of the basal area of the stand $x_{i-1} + G_i$.

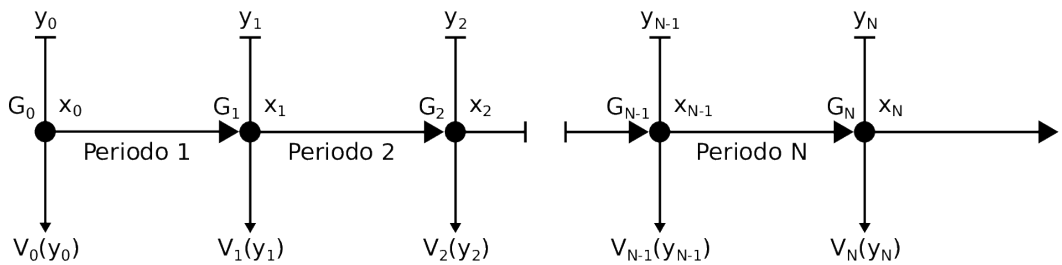

Figure 2: Graphic representation of the DP model. G_i is the growth of the stand from the state x_{i-1} to the start of the period i. x_{i-1} is the state of the stand after the thinning at the start of the period $i-1$. The variable y_{i+1} is the cutting option selected in this period (the percentage of the basal area of the stand to be harvested) from a finite set of cutting options. $V_i(y_i)$ represents the volume of wood harvested from cutting the percentage y_i of the basal area of the stand $x_{i-1}+G_i$

It is clear by looking at constraints (3) and (7), that not all possible harvesting decisions (y_i) lead to feasible solutions. In this regard, two different cases can occur depending on the harvesting decision. The first one, as shown in Subfigure 3 (a), is a feasible selection of a cutting option. In this case, it can be observed that after thinning the stand, the minimum requirements are met, and the growth at the end of the period and the volume harvested are modest but sustainable.

The second case scenario, shown in Figure 3 (b), shows an unfeasible selection of a cutting option. With an intensive thinning of the stand, the volume of wood harvested is higher than in the previous case; however, the remaining trees on the stand are not enough to regenerate it, causing the loss of the stand at the end of the period.

A special case of the first scenario, shown in Figure 3 (c), depicts the option of not harvesting the stand in the current period. In this scenario, there is no volume of wood harvested. However, the growth of the stand is higher in most cases than in the other feasible scenarios. This case could be useful after an intensive, but feasible, thinning of the stand or in conjunction with other methods of harvesting.

Given that the nature of the w_i variable is continuous, and in consequence its search space is infinite, it is common practice to discretize the variable range and allow the cutting options to be taken from a finite (approximate) set in the [0,1] interval. This is achieved by dividing the [0,1] range into subintervals of equal size. Clearly, there is a trade-off in this discretization step. The finer the discretization, the more accurate the solution is. However, the computational cost becomes higher. This issue is investigated in our experimental work.

The previous cases show a given selection of cutting options for a planning horizon. However, to optimize the harvesting of the stand it is necessary to consider all the cutting options on each period to determine the configuration that yields the highest volume of wood harvested, within the current regulations.

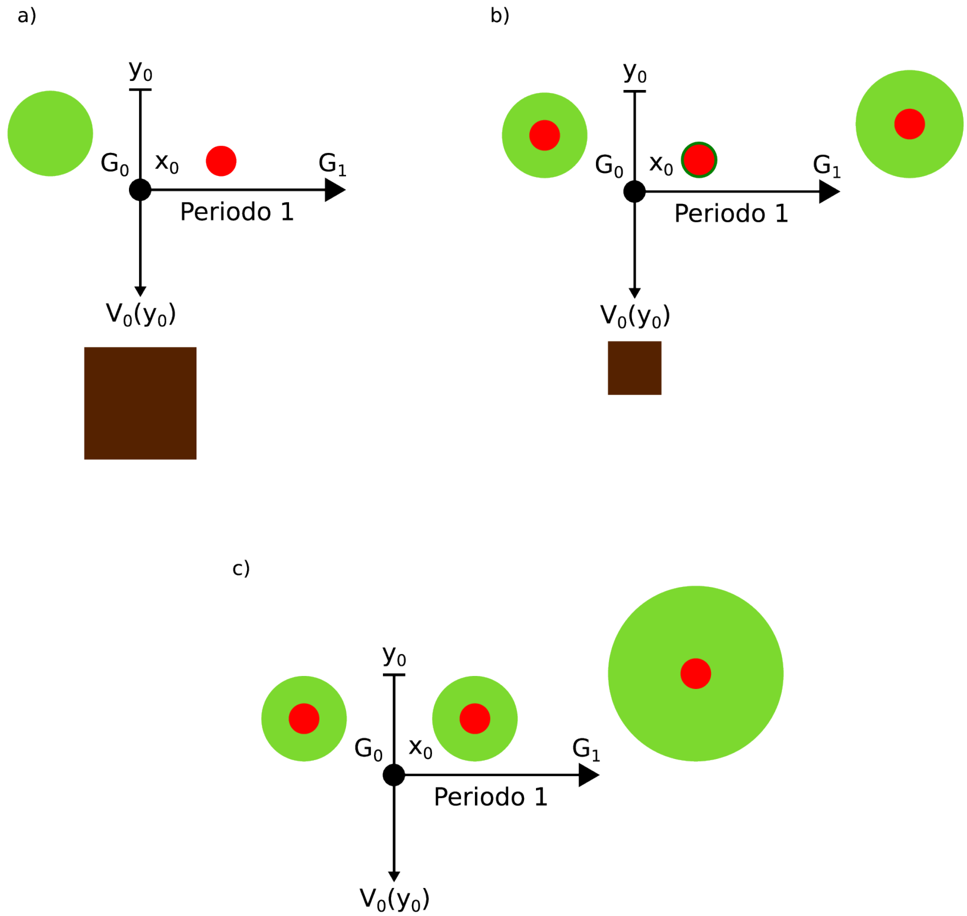

Figure 3: Harvesting options for each period. The green area represents the percentage of the forest that can be harvested. The red circle represents the α vector, or the minimum stock value, and the brown square, the volume of wood harvested using the selected cutting option. Subfigure (a) represents a cutting decision that generates a feasible solution. Subfigure (b) shows a cutting decision that leads to an unfeasible solution. Subfigure (c) represents a cutting decision of not cutting in that period.

3 Implementation of Solution Algorithm

3.1 Dynamic Programming Algorithm

For the implementation of the solution method proposed, the libraries of the freely-available forest growth simulator software BWIN Pro were adaptaed. This was used both to generate the instances of the problem by creating random distributions of trees on a stand of 1 ha, and to simulate the growth of the stand during the periods of the planning horizon.

The pseudocode of the main implementation is shown in Algorithm 1. The algorithm requires an initial stand *root*, the number of periods on which the planning horizon is divided, and the length of each period in years $lengthPeriod$. Note that each search "node" (e.g., root, parent, child) in the algorithm corresponds to a specific stage i and state vector x_i in the DP formulation. Given that the DP formulation is based on a forward recurrence relation, optimal decisions up tp stage i must be known for computing optimal decisions on stage $i + 1$.

Algorithm 1: BFS_DP

input : An initial state root, a list of cut options Y, number of periods $periods$, length of period $lengthPeriod$

output: A final state best with the list of cut options that yields the highest volume of wood

```
1  stack.push(root)
2  while stack ≠ ∅ do
3      parent = stack.pop()
4      for y ∈ Y do
5          child = clone(parent)
6          timber = cut(childStand, y)
7          child.addVolWood(timber)
8          child.grow(lengthPeriod)
9          if child.followsRegulations() then
10             if child.level() < periods then
11                 stack.add(child)
12             end
13             else if child.getVolWood() > best.getVolWood() then
14                 best = child
15             end
16         end
17     end
18 end
19 return best
```

The algorithm starts by adding the *root*, corresponding to the initial state x_0 in stage 0, into a queue structure called *stack* (line [1]). A breath first search (BFS) is iteratively done on until the *stack* is empty (line [2]). At it each iteration a node is retrieved from the top of the queue, and it is called *parent*, see line [3]). This node corresponds to a specific stage i and state x_i of the DP formulation. It is important to remark that each element stores the whole history of the past cut decisions that got from the *root* state to its current state. From this *parent* node, several *child* nodes are generated as follows. Each possible harvesting decision that can be taken at *parent* in stage i, represented by decision y_i in the DP model, leads to a different state in the following stage $i+1$. This new stage and state is associated to a *child* node obtain from this parent. In detail, all possible harvesting choices are explored from the state of the *parent* that lead to feasible solutions that can be reached from it (line [4]). The algorithm creates a *child* node, that is a clone of the *parent* (line [5]), harvest the y percent of the basal area of the *childStand* (line [6]) and add the cut option y to its history, and grows the *child* for a *lengthPeriod* (line [8]). If the *child* is associated with a feasible solution, this is, it satisfies (7), (line [9]), then the algorithm verifies if *child* is not a leaf node (line [10]). If it is not, it is added to the *stack* structure and continues with the search. If it is a leaf node, then it verifies if the volume harvested at the end of the planning period is higher that the obtained in the current *best* solution (line [13]). If it is better, *best* is replaced with *child*. Once the whole *stack* is explored, this is, when the *stack* is empty (line [2]), the algorithm returns the *best* solution found during the exploration (line [19]) and the process stops returning the optimal solution.

An example of the resulting *best* solution for a stand with a size of 1 ha is shown in Figures 4 and 5, where the left size of the figures represent the cutting stages, and the right size the growth of the stand after 8 years, each row represents a period. The individual circles represent trees of the *Pinus cooperi* species. An empty square on the side of the cutting stage, symbolizes that a tree was harvested. The data used for this example considers the regeneration of new individuals. Figure 6 shows on the left, the initial stage of the stand, while the stand on the right shows the final state at the end of the planning horizon, following the process shown in Figures 4 and 5.

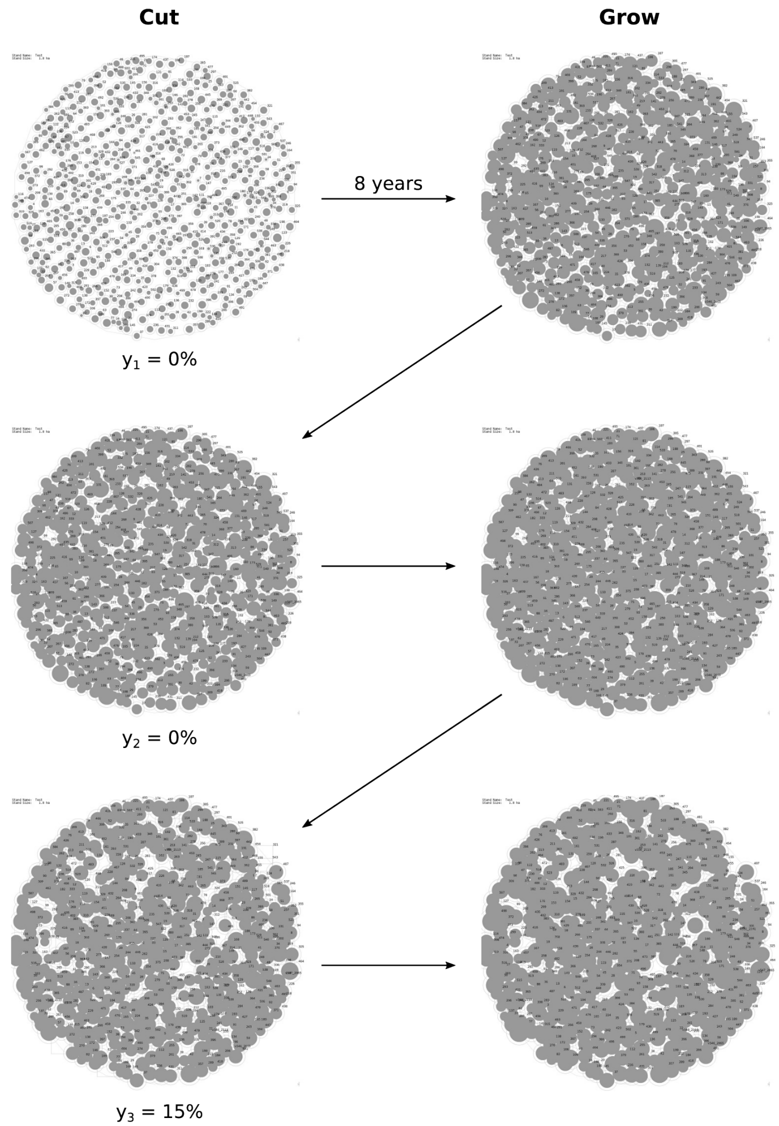

Figure 4: Example of a *best* solution (Periods 1, 2 and 3).

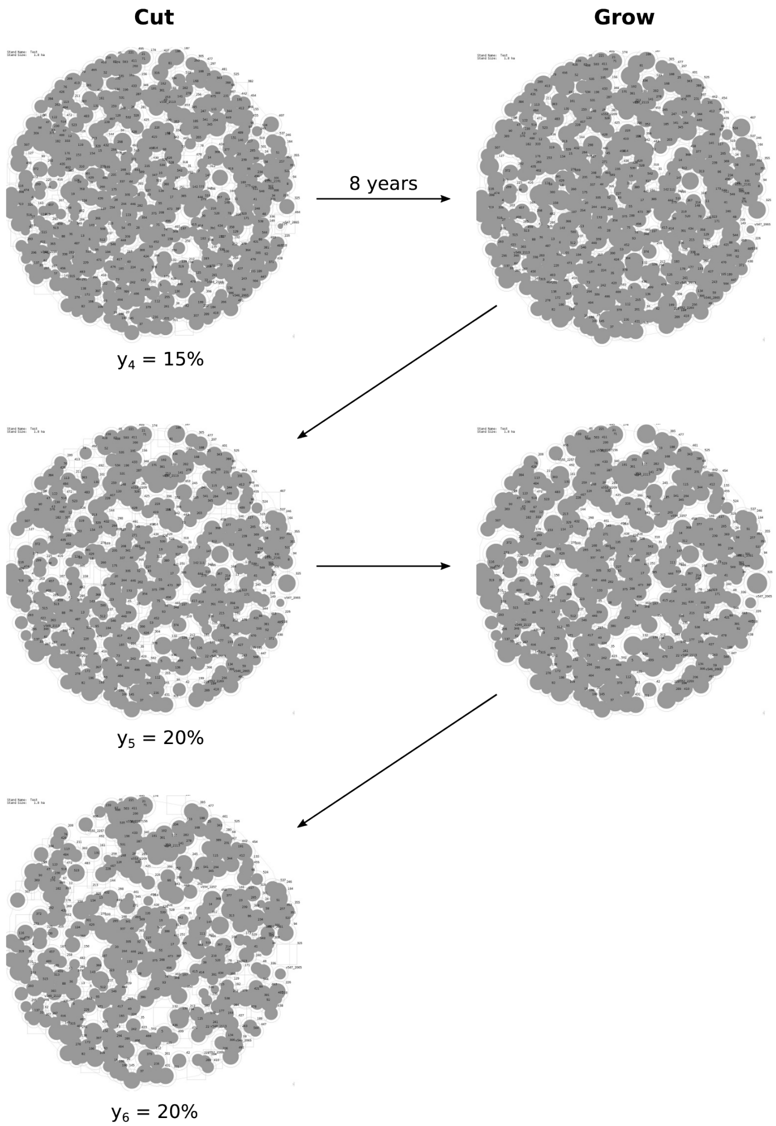

Figure 5: Example of a *best* solution (Periods 4, 5 and 6).

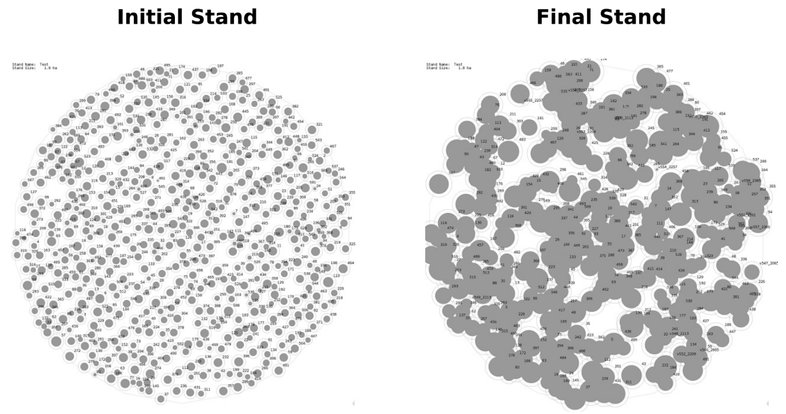

Figure 6: Initial stand and final state of *Best*.

3.2 Forest Growth Simulator

To calculate the growth of the stand during the cut periods in the planning horizon, the TreeGrOSS package (Tree Growth Open Source Software) is used. this package was developed and included in the BWINPro Forest Simulator by Nagel and Schmidt [17]. This package uses a growth model based on single tree information to simulate the evolution of a stand.

As an open-source project, it is possible to audit and modify the source code of the package to adapt it to the particular characteristics of the stand. An example of this is the inclusion of a database containing information of the stand and the species of Las Bayas, in Mexico.

A real stand can be simulated by defining its polygon, and by using the statistical data taken from samples that include information such as the mean basal area of the stand, species, age, dominant height, among others. It is also possible to replicate a stand exactly by defining the coordinates and attributes of each tree that include, but are not limited to, species, age, DBH, height, crown base and width, site index, competition index, etc.

For a random instance, the trees can be distributed in three ways: by random coordinates, as used in this paper, that places the trees in positions were the crowns do not overlap; in raster coordinates, that places them in rows and columns; and in clusters, that generates groups of trees in the stand. Other attributes that are dependent on the region and species being simulated are the inclusion of a regeneration layer, the mortality, and the competition index.

Finally, the documentation recommends not exceeding a planning horizon length of more than 40 years in order to obtain accurate predictions for the growth of the stand.

4 Empirical Work

To assess the performance of our proposed BFS_{DP} solution method with the simulation of a stand at high detail, with the libraries of the open source software ForestSimulator, a series of experiments to be described next are carried out.

The region considered for this example was from the database from the North West of Germany, a stand with the species *Pinus sylvestris* that includes reincorporation values, and from the region of El Salto, in Mexico, with the species *Pinus cooperi*, that does not include reincorporation values. As the libraries of the forest simulator are implemented on the Java programming language, the implementation of the DP method was also made in this language, using the JDK 1.8.0. The experiments were carried out in an Intel i7 CPU at 4.0 GHz. with 32 GB of RAM, under the GNU/Linux Ubuntu 14.4 operating system.

4.1 Determining the range of the cutting options

The first issue to be investigated is whether it is necessary to evaluate the entire range of cutting choices. It is clear that a finer discretization mesh leads to more cutting choices; however, given the additional constraints of minimum cutting patterns, it is not so clear before hand if all possible choices turned out to be feasible. Therefore, some tests to take a closer look of what actual cutting choices turn out to be feasible are carried out.

To this end, a stand with the information of the region of El Salto, in the state of Durango, Mexico, is generated. This stand does not include reincorporation values, such that the range of cutting values is more conservative than if it included them. The regeneration values allow the software to consider the natural appearance of new trees in the stand, in addition to the ones that are already present. As the simulation considers mortality, without reincorporation, the duration of the planning horizon is limited by the lifespan of the trees in the stand. This is because, at some point, the number of trees in the stand does not meet the minimum stock constraint, even when the percentage of harvesting is 0%.

A stand of 1 ha with 500 trees of the *Pinus cooperi* species, with four periods of five years is tested. Four different scenarios are tested as depicted in Table 2.

Table 2: Discretization scenarios for determining the number of cutting options.

Scenario	ΔY (%)	$\|Y\|$
1	25	5
2	10	11
3	5	21
4	2	51

The different amount of granularity obeyed to two reasons, the first one to determine more precisely the limits of the range of the cutting options, and the second to observe the increase of running time of the proposed DP algorithm.

Figure 7 shows the results of this experiment, for four periods of five years each, which gives a planning horizon of twenty years. As the number of periods progress, the value of the cutting options decreases, and the range of options that are evaluated also narrows dramatically, as observed in the box plots.

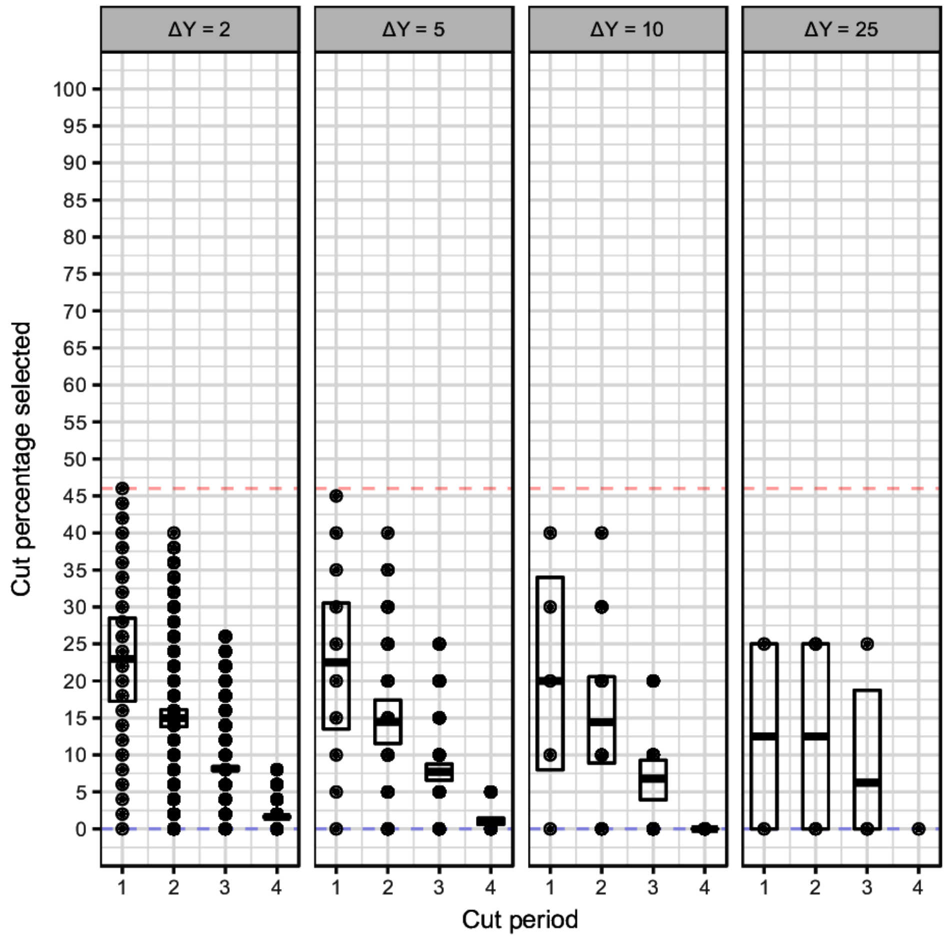

Figure 7: Determining the range of the cutting options for a planning horizon of 20 years, divided in $N = 4$ periods. Each individual subplot shows the results for each scenario. In each subplot, the horizontal axis represents the cutting period, and the vertical axis represents the cutting options considered by the search process during each period. The lower line across the four subsets, represents the lower cutting option selected during the search process, and the top line, represents the higher cutting option selected during the exploration. The box plot on each period shows the range and distribution of the most common cutting options considered.

An important conclusion from this experiment is that, regardless of the discretization size, there is no need to consider cutting choices higher than 50%, therefore in the remaining experiments we only consider the [0-50%] range.

4.2 Determining the number of cutting options

The next issue to investigate is the discretization size of the domain for cutting choices. Recall, that the continuous range for cutting choices is discretized such that the possible cutting choices taken at each stage must belong to a discrete set. The trade-off is evident. That is, the finer the discretization, the more choices we have and the more accurate the optimal solution is. However, the search tree grows with the size of Y thus it is important to assess the trade-off between solution quality and computational effort. As concluded from the previous experiment, we only consider cutting choices in the [0%,50%] range.

In this experiment five scenarios, as depicted in Table 3 (b), are considered. Each scenario considers a cutting selection strategy of trees by age, that is, the trees of the stand are ordered by age in descending order. Then, the algorithm calculates the goal basal area that is equal to $(1 - y_i)$ of the current basal area. After determining this value, and sorting the trees in the stand in descending order, the trees are harvested one at a time until the basal area of the stand is less or equal to the objective basal area.

Table 3: Discretization scenarios for determining the number of cutting options.

Scenario	ΔY (%)	$\lvert Y \rvert$
1	25	3
2	10	6
3	5	11
4	2	26
5	1	51

Figure 8 shows the result of this experiment. The horizontal axis represents the volume of wood harvested (in cubic meters) at the end of the planning horizon.

As can be observed, the volume obtained by dividing the range of cutting options by 1% or by 2% is the same, but the difference in time is higher by one order of magnitud if we divide the range of cutting options in intervals of 1%. Also, it can be observed that dividing by intervals of 5% can be useful for the exploration of other solution techniques with a fair degree of accuracy and a lower time consumption.

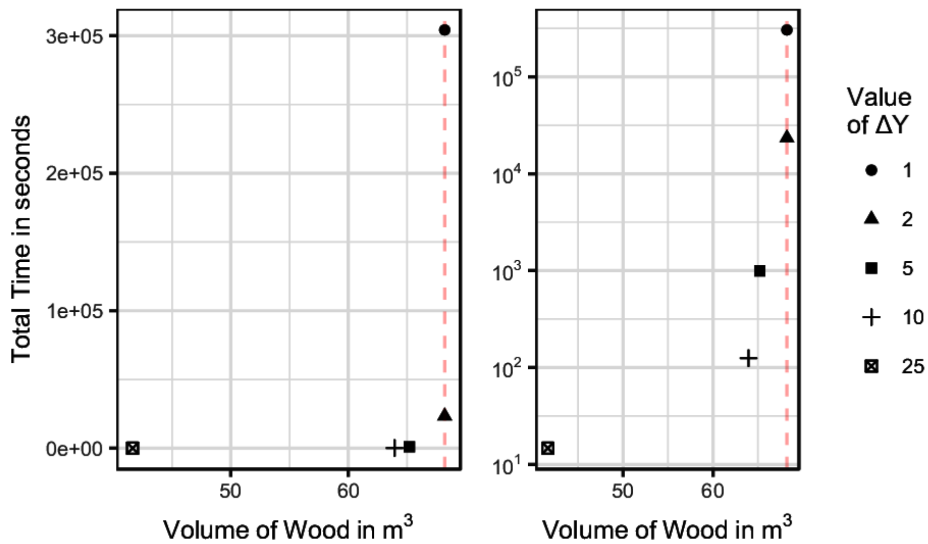

Figure 8: Determining the number of cutting options or ΔY for a planning horizon of 20 years. The horizontal axis represents the volume of wood in cubic meters harvested at the end of the planning horizon. The vertical axis, the time consumed in seconds to find the optimal solution. Each dot represents the size of each interval between 0% and 50%, where a smaller number means a larger number of cutting options. The graph at the left is shown in normal scale, while the graph in the right is in $\log_{10}$ scale. The red dotted line at the right shows the highest volume obtained under these conditions.

4.3 Evaluating alternatives for the selection of trees to be harvested

In the two previous experiments, the selection of the trees to be harvested is made in regard to their age, where the older trees are extracted first. For the third experiment, two additional thinning selection strategies are evaluated. The selection of trees is done by their diameter and by their height. The conditions of the stand are the same as in the other experiments, using the range of cutting choices that was determined in the second experiment. The intervals of the cutting options for these experiments are set at 5 and 2 percent.

Figure 9 shows the result of this experiment. As it can observed, the best selection criteria for the selection of the trees harvested during each cutting stage is choosing those who are taller first. As the consumption of time is the same that using the other two criteria, the height selection criteria in the remaining experiments will be used.

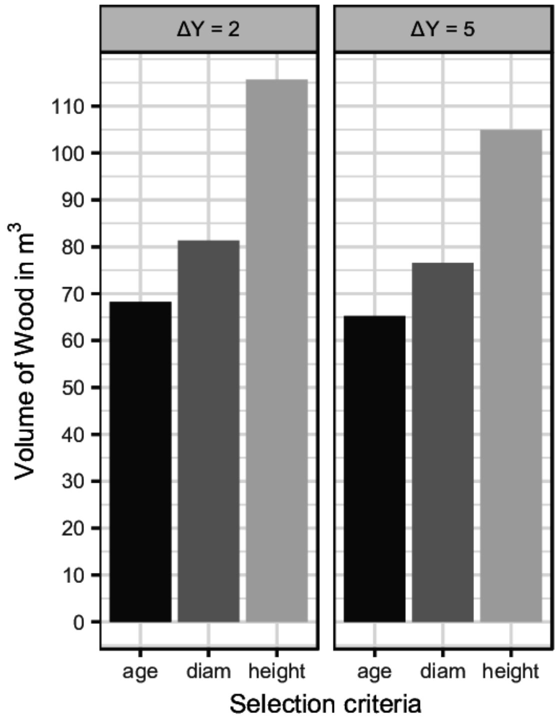

Figure 9: Evaluating alternatives for the selection of trees to be harvested for a planning horizon of 20 years. The horizontal axis represents the three selection criteria considered: age, diameter and height. The subsets represent the size of the intervals of the cutting range used in this experiment. The vertical axis indicates the volume of wood harvested, in cubic meters, at the end of the planning horizon.

It can also be observed, that using a finer discretization ΔY (2 in this case) yields higher volumes of timber for the optimal solutions. This is consistent with the results of the second experiment and it stems from the fact that under the 2% size, there are more choices to be considered in each subproblem through the execution of the DP algorithm. For this experiment we can conclude that the best configuration is to sort the trees by height in descending order, and to use a $\Delta Y = 2$ as the interval between cutting options.

4.4 Evaluating the length of the periods in the planning horizon

In the previous experiments the time span of each harvesting period was fixed at five years, as this is an interval value often used in yield tables in practice. Because this is somehow an arbitrary decision, it is interesting to investigate the effect of having harvesting periods of different length. To this end, in this last experiment different planning horizons and time periods are assessed. Particularly, three planning horizons of 12, 24, and 48 years, each one divided into 2, 3, 4, and 6 harvesting periods through the planning horizon, are considered.

For this experiment, the data from a simulated forest in the West Forest in Germany is used. This includes the incorporation of new trees during the simulation. Also, in this experiment the discretization size of $\Delta Y = 5$ is considered. It is important to highlight that this kind of experiment cannot be done with the forest stands from El Salto because instances with periods longer than 25 years cannot be simulated with the available data.

Table 4 depicts the results of this experiment for the different scenarios. For instance, in the first row, we show the results for a planning horizon of 12 years, where there are six harvesting periods of length of 2 years each.

Table 4: Best cut options for each planning horizon with different cut periods and total volume of wood harvested. Column "Planning" shows the length of the planning horizon. Column "Periods" shows the number of time periods considered for each planning horizon. Column "Years" indicates the lenght of each harvesting period. Finaly, columns "Cut Options" and "Wood Volume" show the optimal solution (cutting pattern) and its corresponding optimal solution value, respectively, found by the BFS_DP algorithm.

Planning	Periods	Years	Cut Options	Wood Volume
12	6	2	0, 0, 10, 15, 30, 30	1192.8273
	4	3	0, 0, 25, 35	522.1944
	3	4	0, 15, 40	333.0745
	2	6	5, 45	185.9485
24	6	4	5, 0, 5, 15, 25, 25	2136.5076
	4	6	0, 0, 25, 35	1077.2481
	3	8	0, 15, 40	677.3736
	2	12	0, 50	374.4359
48	6	8	0, 0, 15, 15, 20, 20	5329.3579
	4	12	0, 5, 30, 30	2910.9219
	3	16	0, 20, 40	1836.3285
	2	24	0, 50	971.6463

In this table, it can be observed that for all planning horizons, when the number of cutting periods is larger, the volume collected at the end is also higher with conservative cutting options. On the other hand, when the number of cutting phases is shorter, the volume harvested is lower and the cutting options are more intensive.

Figure 10 summarizes these results, where it can be visually seen that increasing the number of periods for each planning horizon improves the volume of wood collected without violating the minimum stock constraint.

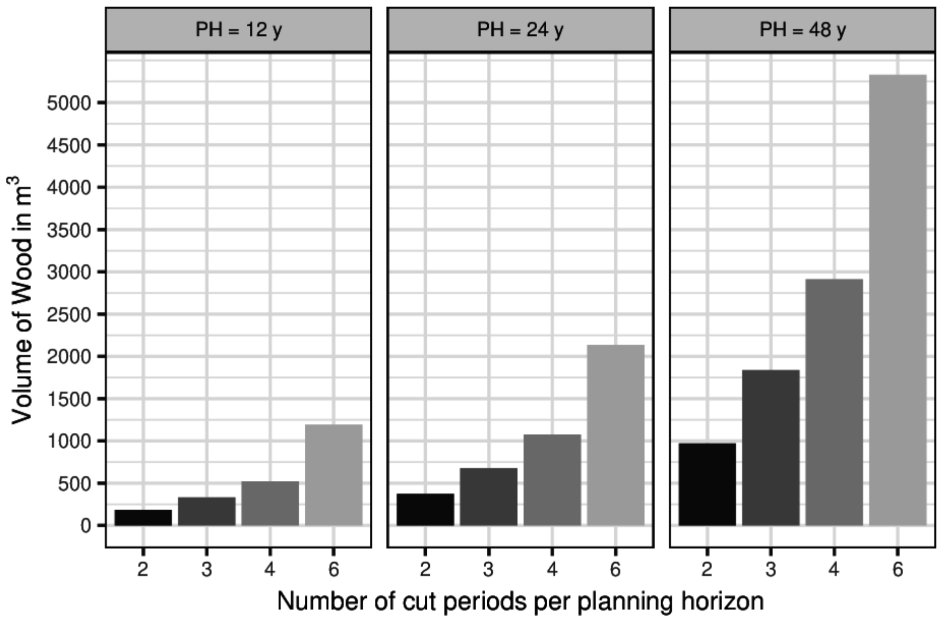

Figure 10: Evaluating the length of the periods for three different planning horizons of 12, 24 and 48 years, each one divided in $N = \{2, 3, 4, 6\}$ periods. The vertical axis indicates the volume of wood harvested, in cubic meters, at the end of the planning horizon.

Table 5 shows the time in seconds at which the algorithm found the best solution and the time in seconds that the algorithm took to finish the exploration of the search tree. The times obtained increase rapidly when the number of cutting phases is larger. While increasing the duration of the planning horizon does not seem to grow as fast.

Figure 11 shows in square-root scale, the time consumed by each number of periods per planning horizon. As we can observe, the duration of the planning horizon had less impact in the time consumption than the number of periods on which it is divided.

Table 5: Time on which the best solution was found. Column "Planning" shows the length of the planning horizon. Column "Periods" shows the number of time periods considered for each planning horizon. Column "Time Found (sec.)" shows the time in seconds and in hours at which the algorithm found the best solution. Finally, column "Total Time (sec.)" shows the time in seconds that the algorithm took to finish the exploration of the search tree.

Planning	Periods	Time Found (sec.)	Total Time (sec.)
12	6	10892.82	35388.46
	4	285.84	848.79
	3	44.19	139.87
	2	6.90	15.83
24	6	24564.94	53501.67
	4	421.07	1449.32
	3	54.73	192.13
	2	5.90	17.48
48	6	23811.58	65707.41
	4	753.66	2433.13
	3	92.35	318.14
	2	7.30	26.83

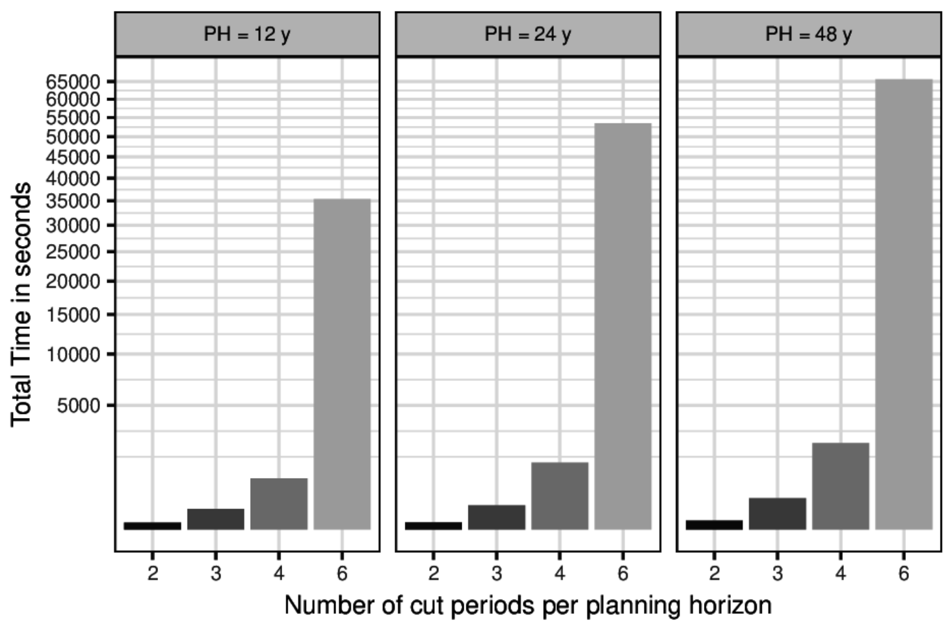

Figure 11: Evaluating the length of the periods for three different planning horizons of 12, 24 and 48 years, each one divided in $N = \{2, 3, 4, 6\}$ periods. The vertical axis indicates the total time consumed by the algorithm to find the optimal solution in seconds in square-root scale.

5 Conclusions

The synergy of both, the DP model and the forest growth simulator software, gave us the capability of exploring alternatives to the current policies for treatments, such as the intensity of thinning and the time span between periods to maximize the volume of wood, while avoiding the overexploitation of the stand.

For the experiment where it was sought to determine the minimum range of cutting options to consider, in order to reduce the amount of computational time consumed, it was found that the range was strongly dependent on the current minimum stock constraint, thus, different instances require a previous analysis to determine this value. In return, at least for the instance solved during the experimentation, it is possible to reduce this range by 50%.

When analyzing the ΔY or size of the interval that yielded the highest volume of wood in the instance problem , it was found that the best configuration was $\Delta Y = 2$, as it obtained the same volume of timber as with a configuration of $\Delta Y = 1$, with a lower consumption in time by an order of magnitude. $\Delta Y = 5$ offers a good compromise between computational time and volume of wood harvested, which could be useful for larger instances or as a tool for exploratory analysis of future strategies.

For the third experiment, where we wanted to determine the best criteria to select the trees to be harvested during each thinning period, the results showed that the best selection strategy of the three that we tested, was the selection by Height, with no impact on the computational time, but with a gain in volume harvested at the end of the planning horizon of up to 65%.

When the best number of periods in the planning horizon and their duration were analyzed, the results of the experiment showed that periods of shorter duration yielded the highest volume of wood, up to six times compared with the lower number of periods considered in the experiment. But, as in the case of the values for ΔY, shorter periods require a higher consumption of computational time, as the planning horizon is divided in a larger number of them.

This method also serves as a tool to explore alternative strategies to those previously established by tradition or by alternative methods, with the confidence that the results obtained are reasonably close to reality, thanks to the forest growth simulator module.

Future works following the path of this project will include the inclusion of additional species, for a heterogeneous stand. Another line of research is to consider multiple stands with adjacency constraints. Another area of opportunity is the study of models that take into account the costs of the infrastructure deployed during each partial thinning.

Acknowledgments

The first author was supported by a posdoctoral fellowship from the Teaching Development Program (PRODEP) [511-6/17-2607]. The fourth author would like to thank Tecnologico de Monterrey Research Group in Optimization and Data Science.

References

[1] E. L. Amidon and G. S. Akin. Dynamic programming to determine optimum levels of growing stock. *Forest Science*, 14(3):287–291, 1968.

[2] G. J. Arthaud and W. D. Klemperer. Optimizing high and low thinnings in loblolly pine with dynamic programming. *Canadian Journal of Forest Research*, 18(9):1118–1122, 1988.

[3] P. Asante, G. W. Armstrong, and W. L. Adamowicz. Carbon sequestration and the optimal forest harvest decision: a dynamic programming approach considering biomass and dead organic matter. *Journal of Forest Economics*, 17(1):3–17, 2011.

[4] R. Bellman. The theory of dynamic programming. Technical report P-550, The RAND Corporation, Santa Monica, 1954.

[5] P. Bettinger, D. Graetz, and J. Sessions. A density-dependent stand-level optimization approach for deriving management prescriptions for interior northwest (USA) landscapes. *Forest Ecology and Management*, 217(2):171–186, 2005.

[6] J. D. Brodie, D. M. Adams, and C. Kao. Analysis of economic impacts on thinning and rotation for Douglas-fir, using dynamic programming. *Forest Science*, 24(4):513–522, 1978.

[7] J. D. Brodie and C. Kao. Optimizing thinning in Douglas-fir with three-descriptor dynamic programming to account for accelerated diameter growth. *Forest Science*, 25(4):665–672, 1979.

[8] V. Brukas and J. D. Brodie. Economic optimisation of silvicultural regimes for Scots pine using dynamic programming. *Baltic Forestry*, 5(1):28–34, 1999.

[9] D. E. Chappelle and T. C. Nelson. Estimation of optimal stocking levels and rotation ages of loblolly pine. *Forest Science*, 10(4):471–502, 1964.

[10] C. M. Chen, D. W. Rose, and R. A. Leary. Derivation of optimal stand density over time – A discrete stage, continuous state dynamic programming solution. *Forest Science*, 26(2):217–227, 1980.

[11] L. Diaz-Balteiro and L. C. E. Rodriguez. Optimal rotations on *Eucalyptus* plantations including carbon sequestration – A comparison of results in Brazil and Spain. *Forest Ecology and Management*, 229(1–3):247–258, 2006.

[12] D. H. Graetz, J. Sessions, and S. L. Garman. Using stand-level optimization to reduce crown fire hazard. *Landscape and Urban Planning*, 80(3):312–319, 2007.

[13] R. G. Haight. Evaluating the efficiency of even-aged and uneven-aged stand management. *Forest Science*, 33(1):116–134, 1987.

[14] R. G. Haight, J. D. Brodie, and W. G. Dahms. A dynamic programming algorithm for optimization of lodgepole pine management. *Forest Science*, 31(2):321–330, 1985.

[15] C. Kao and J. D. Brodie. Determination of optimal thinning entry interval using dynamic programming [*Pseudotsuga menziesii*]. *Forest Science*, 25(4):672–674, 1979.

[16] M. Meneses and B. Olivares. Estrategia óptima de corta de nivel de rodal en plantaciones de *Pinus radiata*. *Bosque*, 7(1):3–8, 1986.

[17] J. Nagel and M. Schmidt. The silvicultural decision support system BWINPro. In H. Hasenauer, editor, *Sustainable forest management: Growth models for Europe*, pages 59–63. Springer, Berlin, Germany, 2006.

[18] G. L. Paredes V. and J. D. Brodie. Efficient specification and solution of the even-aged rotation and thinning problem. *Forest Science*, 33(1):14–29, 1987.

[19] N. A. Ribeiro, P. Surový, and A. Yoshimoto. Optimal regeneration regime under continuous crown cover requirements in cork oak woodlands. *FORMATH*, 11:83–102, 2012.

[20] O. Tahvonen and J. Rämö. Optimality of continuous cover vs. clear-cut regimes in managing forest resources. *Canadian Journal of Forest Research*, 46(7):891–901, 2016.

[21] M. C. Wimberly and B. B. Bare. Distance-dependent and distance-independent models of Douglas-fir and western hemlock basal area growth following silvicultural treatment. *Forest Ecology and Management*, 89(1–3):1–11, 1996.

[22] A. Yoshimoto, P. Asante, and M. Konoshima. Stand-level forest management planning approaches. *Current Forestry Reports*, 2(3):163–176, 2016.

[23] A. Yoshimoto, R. G. Haight, and J. D. Brodie. A comparison of the pattern search algorithm and the modified PATH algorithm for optimizing an individual tree model. *Forest Science*, 36(2):394–412, 1990.

[24] A. Yoshimoto and R. Marušák. Evaluation of carbon sequestration and thinning regimes within the optimization framework for forest stand management. *European Journal of Forest Research*, 126(2):315–329, 2007.

[25] A. Yoshimoto, G. L. Paredes V., and J. D. Brodie. Efficient optimization of an individual tree growth model. *FORMATH*, 15:20–32, 2016.

A Notation

Table 6 contains the notation and description of the abbreviations of the optimization methods mentioned in this paper.

Table 6: Description of the notation regarding solution methods used in this work. Column "Abbreviature" is the abbreviature of the optimization method and column "Description" the meaning of the abbreviature.

Abbreviature	Description
DP	Dynamic Programming
Stoch. DP	Stochastic Dynamic Programming
BFS	Breadth-first Search
MILP	Mixed-Integer Linear Programming
Heur.	Heuristic
GA	Genetic algorithm
TS	Tabu Search
PSO	Particle Swarm Optimization
DE	Differential Evolution
EA	Evolutionary Algorithm