Instituto Superior
de Engenharia
Politécnico de Coimbra

DEPARTMENT OF SYSTEMS AND COMPUTER ENGINEERING

Simulator For Forest Management Alternatives

Project Report to fulfill the Master’s degree in Informatics Engineering

Specialization Area of Intelligent Data Analysis

Author

Isabel Catarina Ramos Castro

Advisors

Mateus Daniel A. Mendes

José Torres Farinha

Supervisors

Beatriz Fidalgo
Raúl Salas-González

INSTITUTO POLITÉCNICO
DE COIMBRA

INSTITUTO SUPERIOR
DE ENGENHARIA
DE COIMBRA

Coimbra, December 2024

RESUMO

As florestas fornecem serviços essenciais para o ecossistema, mas a sua gestão sustentável permanece um desafio. A otimização do volume de madeira extraída e a avaliação de diferentes estratégias de gestão exigem a simulação do futuro desenvolvimento dos povoamentos florestais. Para auxiliar neste processo, os gestores florestais necessitam de ferramentas que lhes permitam tomar decisões informadas e baseadas em dados, uma gestão que equilibre metas imediatas com a sustentabilidade a longo prazo.

Este trabalho aborda esses desafios através do desenvolvimento de um simulador web de gestão de alternativas florestais, dotado de um algoritmo genético adaptado (NSGA-II). O simulador está disponível na plataforma "Floresta Digital". Para garantir a sua integração bem-sucedida, o processo de desenvolvimento foi dividido em duas etapas: primeiro, foram realizadas melhorias no website existente e, em seguida, foi criado e implementado o simulador.

O simulador permite que os utilizadores explorem diversas opções de gestão florestal, avaliando o equilíbrio entre a extração de madeira e a variação do volume ao longo do tempo, apresentando múltiplas alternativas de gestão. A modificação do algoritmo possibilitou a criação de um maior número de possíveis combinações de povoamentos, satisfazendo as restrições do problema e indo além dos métodos tradicionais. Para avaliar o desempenho do NSGA-II modificado, foram aplicados vários indicadores de desempenho e parâmetros diferentes, comparando-os com a versão padrão. Os resultados indicam que o algoritmo otimizado gera um maior número de soluções válidas sem aumentar significativamente o tempo de execução, proporcionando um melhor desempenho em termos de diversidade de soluções e eficiência computacional.

O simulador fornece dados de uma floresta em Coimbra, Portugal, para estudos experimentais, mas também suporta conjuntos de dados de outras localidades para problemas de otimização semelhantes. A criação do simulador segundo um formato Web permite aumentar a acessibilidade para os profissionais da área florestal. No entanto, o simulador apresenta algumas limitações, particularmente no que diz respeito à introdução manual de dados e à definição de objetivos por parte do utilizador. Estas limitações proporcionam oportunidades para melhorias futuras. Este trabalho contribui para a gestão sustentável das florestas ao fornecer uma ferramenta prática de simulação para apoiar a tomada de decisão pelos gestores florestais.

Palavras-chave: Gestão Florestal, Simuladores Florestais, Otimização Multi-Objectivo, Algoritmos Genéticos, NSGA-II

ABSTRACT

Forests provide essential ecosystem services, but their sustainable management presents a complex challenge. Achieving objectives such as timber harvesting optimisation require simulating the future development of forest stands and evaluating various management strategies. To support this, forest managers need access to decision support tools that enable informed, data-driven choices, ensuring that management practices balance immediate goals with long-term sustainability.

This work addresses these challenges by developing a web-based forest management simulator integrated with an adapted genetic algorithm (NSGA-II). The simulator is available on the "Floresta Digital" platform. To successfully integrate the simulator, the development process was divided into two stages: first, improvements to the existing website, followed by the creation and implementation of the simulator itself.

The simulator allows stakeholders to explore various forest management options while assessing the trade-off between timber harvesting and volume variation over time, generating multiple management alternatives. By enhancing the algorithm to generate a wider range of valid stand combinations, this modification not only satisfies problem constraints but also outperforms traditional methods. To evaluate the performance of the custom NSGA-II, several performance metrics and different parameters were applied and compared to the standard version. The results suggest that the custom NSGA-II generates a larger number of valid solutions without a significant increase in runtime, offering superior performance in terms of solution diversity and computational efficiency.

The simulator provides data from a forest in Coimbra, Portugal, for experimental research, but also supports datasets from other locations for similar optimisation problems. Its web-based design ensures accessibility for non-experts, however, limitations like manual data entry and user-defined objectives highlight opportunities for enhancements. This study strengthens sustainable forest management by providing a practical simulation tool to aid decision-making for forest stakeholders.

Keywords: Forest Management, Forest Simulators, Multi-Objective Optimization, Genetic Algorithms, NSGA-II

EPIGRAPH

"You can't just turn on creativity like a faucet. You have to be in the right mood.
What mood is that?
Last-minute panic."

– Calvin and Hobbes by Bill Watterson

DEDICATION

Dedico este trabalho ao meu pai e ao meu avô,
madeireiros de profissão,
que, apesar das suas breves passagens por este mundo,
deixaram em mim uma valiosa herança:
o interesse pela floresta.

Sem o saberem, foram a inspiração
que me guiou na escolha deste tema.

ACKNOWLEDGEMENTS

Around six years ago, I made the difficult decision to enrol at ISEC in the course of Computer Engineering. It was a choice filled with fears, but also great expectations. Today, after many sleepless nights and countless moments of joy, I have no doubt that I would make the same choice again.

First, I would like to express my deep gratitude to my supervisors, Professor Mateus Mendes and Professor Torres Farinha. Their mentorship extended beyond the curriculum; it inspired a passion for learning. They generously shared their vast knowledge, and the detours during our meetings were often as valuable as the planned discussions, strengthening my respect for the entire ISEC faculty. Those meetings were always a highlight of my week.

I also want to extend my heartfelt thanks to Professor Raúl Salas-Gonzalez and Professor Beatriz Fidalgo from ESAC. My studies in forest management occupied a significant portion of this past year and their patience and dedication were absolutely essential. As a small token of my appreciation, I leave them with a completed *Floresta Digital*, now even more beautiful —- a true sight for sore eyes.

To my friends, my sincere thanks. You were the ones who listened to my complaints, supported me, and encouraged me to keep going. The trust you have in me is unshakeable.

And last but certainly not least, I am also eternally grateful to my mother, who always encouraged me to follow my dreams, no matter how difficult they seemed. For a whole year, she patiently listened to me talk endlessly about this project that I loved so much, never once telling me to be quiet, and was always by my side.

Até já, Coimbra.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF ILLUSTRATIONS

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
AIGP	Integrated Landscape Management Areas
ANN	Artificial Neural Networks
AUC	Area Under the Curve
BBN	Bayesian Belief Networks
CD/CD	Continuous Integration/ Continuous Deployment
CNN	Convolutional Neural Networks
DP	Dynamic Programming
ESAC	Escola Superior Agrária de Coimbra
GA	Genetic Algorithm
GBDT	Gradient Boosting Decision Trees
GDP	Gross Domestic Product
GIS	Geographic Information System
IDE	Integrated Development Environment
ISEC	Instituto Superior de Engenharia de Coimbra
LiDAR	Light Detection and Ranging
LP	Linear Programming
MDP	Markov Decision Process
ML	Machine Learning
MOGA	Multiple Objective Genetic Algorithm
NPGA	Niched Pareto Genetic Algorithm
NSGA-II	Non-Dominated Sorting Genetic Algorithm II
ORM	Object-Relational Mapping
RF	Random Forest
RL	Reinforcement Learning
SA	Simulated Annealing
RNN	Recurrent Neural Networks
SVM	Support Vector Machines
WSGI	Web Server Gateway Interface

LIST OF SYMBOLS

C	Number of compositions
CO_2	Carbon dioxide
I	Number of species in each composition
j	Number of harvest periods
N	Number of stands
P	Number of species
$P_{proportion}$	Ratio of species *p* in stand *n*
S_{area}	Proportion of usable area in stand *n*
S_{total}	Standard deviation value during the planning horizon for all stands *N*
T	Number of harvest periods
V_{total}	Volume harvested during the planning horizon for all stands *N*
V_{xtc}	Timber volume harvested during period *t* for species *x* of stand *n*
$\bar{x}$	Mean of the sample for period *t*
v_{stand}	Volume of the stand for period *t*
$v_{thinnings}$	Volume obtained from thinning
X	Stand's timber volume for period *t*

1 INTRODUCTION

1.1 Context to Forest Management

Forests are complex systems that provide a wide range of ecosystem services that contribute to environmental sustainability and resilience. However, the quantity and diversity of these services depend heavily on the type of forest management applied and the initial conditions of the landscape.

In Portugal, particularly in the central region, the forest landscapes are mainly composed of planted forests owned by many small non-industrial private landowners whose main objective is wood production [1, 2].

The extensive use of maritime pine and eucalyptus, along with agricultural abandonment and lack of active forest management, has led to large areas of forest monocultures and scrublands. These conditions, combined with orographic and climatic factors typical of the region, as well as climate change, have resulted in widespread forest fires and the spread of invasive wood species. This has caused the loss of ecological and economic value of forest areas and their degradation, significantly altering the landscape and compromising sustainability.

In response to this concern, Portugal has adopted the recommendation of several international bodies and policy makers from the European Union to use a landscape approach to forest management [3–5]. This approach aims to better tailor policy measures and forest management to meet the expectations of local stakeholders and communities and to promote the resilience of forested landscapes [5, 6].

A new policy instrument known as AIGP (Integrated Landscape Management Areas) has emerged to promote forest management at the landscape level, proposing a multifunctional model for forest spatial planning. This model is based on functional zoning, which divides the territory into three categories: (1) the resilience structure aims to reduce fire hazards; (2) the ecological structure to ensure the environmental services of the landscape and the conservation of natural areas and biodiversity; and (3) the remaining area that does not have any designation but will be called a matrix in this work, where wood production and other forest products with economic value can be pursued to meet the needs of forest landowners.

The first two structures are designed to adhere to the spatial environmental restrictions specified in the hierarchical planning system. In contrast, in the matrix, the use of a wide range of forest management models and silvicultural options is encouraged to promote landscape heterogeneity and structural diversity. Diversifying management strategies can improve the ecological and economic resilience of the landscape, helping to reduce the likelihood of catastrophic events and the economic dependence on a minimal range of forest products [7].

Management strategies should integrate silvicultural practices to improve forest resilience to climate change. These practices, among other adaptation measures, include species diversification, the establishment of mixed stands, the promotion of natural regeneration, adjustments to harvesting ages, and the implementation of shelterwood and selective cuts [8–12]. The diversification of silvicultural practices makes the process of selecting management alternatives significantly more complex, requiring a careful evaluation of ecological, economic, and operational trade-offs.

Furthermore, historical, political, and social factors significantly influence the effectiveness of forest adaptation strategies, often limiting the universal applicability of findings from global studies in addressing global change [13]. Therefore, it is essential to develop locally tailored adaptation measures that account for specific regional contexts and stakeholder needs, and the long-term sustainability of forest ecosystems. This adds another layer of complexity to the already intricate challenge of designing forest management compositions and drafting management plans [14, 15]. Many authors emphasise the importance of the learning process involved, highlighting its fundamental contribution to the quality and feasibility of the solutions developed [16, 17].

The specific social context in Portugal amplifies the importance of the learning process in forest management. Exploring new approaches to adaptive silviculture for forest stands and designing comprehensive management plans for areas with multiple forest owners are essential. The inherent complexity of the situation makes it difficult for forest owners and decision-makers to clearly define their preferences without understanding how different management alternatives align with their objectives and potential outcomes.

1.2 Research Motivation

Forest management is a multifaceted challenge that requires balancing a variety of objectives, such as maximising timber yield, preserving biodiversity, and ensuring environmental sustainability. Traditional optimisation techniques, such as linear and dynamic programming, have been used to address these challenges, with notable studies by Hoganson *et al.* [18] and Borges *et al.* [19].

However, these methods often face limitations when applied to large, complex, and multi-objective problems [20], especially as the scale and complexity of forest management increases. In these cases, the computational cost of traditional approaches can become prohibitive.

In order to address these limitations, heuristic approaches have been employed as a practical alternative, offering near-optimal solutions with manageable computational costs [20, 21]. Among these, Genetic Algorithms (GAs) have emerged as a powerful and widely used tool to solve complex, large-scale, multi-objective optimisation problems [22]. GAs utilise evolutionary principles to improve the populations of potential solutions over multiple generations [21]. In contrast to single-objective methods, which consolidate various objectives into a single metric [23], GAs generate a set of solutions that independently balance multiple objectives. As a result, this approach is well suited to scenarios where the optimal solution may be represented by more than one alternative.

A key development in this field is the Non-Dominated Sorting Genetic Algorithm II (NSGA-II), which was introduced by [24]. The effectiveness of NSGA-II in producing a range of Pareto-optimal solutions has been demonstrated, allowing decision-makers to examine a variety of trade-offs rather than a single, potentially biased solution. The core mechanisms of the algorithm, namely the non-dominated sorting and crowding distance, assist in the preservation of diversity within the set of solutions, thus ensuring a wide selection of high-quality results [24].

However, despite its proven effectiveness, such advanced optimisation techniques are often inaccessible to forest managers due to the lack of open-source and user-friendly platforms.

This research aims to fill a gap in forest management by offering a versatile and open source tool: a web-based simulator for management alternatives. This simulator enables decision-makers to assess the influence of a variety of management strategies across defined planning periods in relation to timber volume, without previous knowledge of optimisation algorithms or the need to define a mathematical problem.

For this work, a customised version of NSGA-II has also been developed and integrated with the simulator, featuring a mutation operator designed to enhance the diversity and quality of the solution, in accordance with the simulator requirements.

The entire process is accessible through the website `www.floresta.digital.esac.pt`, which not only hosts this simulator, but also offers a range of additional forest tools, providing a comprehensive platform dedicated to sustainable forest management.

This project resulted in the publication of the article "Optimising Forest Management Using Multi-Objective Genetic Algorithms" in the journal *Sustainability MDPI* (December 2024) [25].

1.3 Methodology

This project was carried out in two separate phases. Initially, the first phase concentrated on examining and refining the existing website platform associated with the project. This phase included a series of enhancements designed to improve the functionality and accessibility of the platform. Subsequently, the second phase was dedicated to the creation of a forest management simulator. The simulator has been designed with a multistep format in order to facilitate an intuitive and straightforward user experience.

In relation to the simulator, it has been constructed from two integrated modules. The first module enables users to import forest stands and simulate potential management options for individual stands, allowing them to test different scenarios. The second module employs a custom NSGA-II algorithm designed to generate more diverse solutions tailored to the context of the forest management problem.

The performance of the algorithm was evaluated through a practical case study, comparing the customised NSGA-II with the standard NSGA-II across various parameter settings. The study aimed to optimise two primary objectives: maximising timber harvest volume and minimising standard deviation in harvest volumes throughout the planning horizon, while ensuring that all constraints were met. The results demonstrate the algorithm's capability to offer forest managers a very diverse set of solutions, helping to get better support for decision-making and trade-off analysis.

1.4 Outline of the Work

This work is organised into eight chapters, each focused on a key aspect of the research. The structure of the document is as follows.

Chapter 1 introduces the research by summarizing the study's objectives, methodologies, and document structure. This chapter sets the context and delineates the main goals.

A comprehensive review of the literature concerning multi-objective optimisation techniques is presented in Chapter 2. This section explores the application of these techniques in contexts such as forest management, emphasising the evolution of these methods and their significance in addressing optimisation problems.

Chapter 3 concentrates on the technical execution of the research, detailing the tools and libraries utilised for constructing the simulator's web platform.

The design and architecture of the simulator are outlined in Chapter 4. This chapter explores the development of the system, discusses initial challenges, identifies areas for improvement, and describes the refinements made to the user experience.

Chapter 5 focusses on the initial module of the simulator, providing an in-depth analysis of its requirements, the design of its data structure, and the workflow of the multistep form that guides the simulation process.

The incorporation of a multi-objective optimisation algorithm into the simulator is discussed in Chapter 6. This chapter provides an overview of the context of forest management and presents a mathematical formulation of the optimisation problem. It also outlines the strategy used to adapt the algorithm to the specific requirements of the study. A case study is used to illustrate the algorithm's performance, followed by an analysis of the results.

Chapter 7 presents a critical analysis of the completed work, evaluating its advantages, limitations, and primary contributions.

In conclusion, Chapter 8 provides a summary of key findings and offers recommendations for future research directions.

2 RELATED WORK

This chapter presents a detailed examination of current research in forest management, with a specific attention to the evolution of methodologies employed in forest ecosystem management. The objective is to identify current trends and address existing literature gaps to support decision-making based on simulation tools in the field.

The following sections summarize the evolution of forest management strategies, starting with traditional methods for optimizing forestry practices. It then examines heuristic approaches, such as Genetic Algorithms, tailored for multi-objective optimisation. Next, the application of AI techniques in forest related contexts is reviewed. The chapter ends with a summary of key findings from the literature.

2.1 Evolution of Methodologies for Forest Management

In forest management, effective decision making is key. Planning serves as a crucial tool to facilitate decision processes within forests and aims to empower decision makers with the information and strategies required to achieve the balance between inputs and outputs [26]. This balance helps to find the best goals set for sustainable management of forested areas, these goals range from ecological and economic considerations to social and recreational aspects [27]. In real life, forests are expected to generate reasonable incomes, but also promote conservation and recreational aspects, making them versatile enough to serve multiple purposes simultaneously.

Over the years, the landscape of forest management has evolved significantly. Reliable assessment of forest productivity is essential for efficient forest management. However, productive forest management is aided by the efficient generation of precise models of forest productivity. So in its earlier stages, forest management relied predominantly on mathematical techniques to devise strategies to optimise forest resource use. These studies were instrumental in the first steps towards more sophisticated tools for predicting the future dynamics of forests [28].

However, as technology advanced and new challenges such as climate change emerged, forest management has seen a shift towards the incorporation of more computationally based techniques. These techniques cannot guarantee finding only one optimal solution to a planning problem [29], however, given the uncertainty and the fact that forest management is dynamic in time and space, the very notion of trying to find only one solution may be a limitation of those mathematical techniques.

It is crucial to keep in mind that environmental uncertainty is a factor in long-term sustainable forest management [30].

Computational based techniques, in particular the use of Machine Learning algorithms, have helped to address challenges related to models performance, scalability, and evaluation of properties for complex cases. The main goal of both models is to help the user gain insight into changes that affect forest productivity over time [31].

2.2 Exact Approaches in Forest Management

Societies are heavily dependent on forest ecosystems, as they play a crucial role in providing essential services. More than half of the global gross domestic product (GDP) is estimated to be substantially dependent on ecosystem services, notably those provided by forests [32].

However, the way wood is being cultivated and cut is still based on rules of thumb from the 1900s. Despite the need to pay more attention to forests and their ecosystems, there is still difficulty in using several methods that have been studied and proven to help optimise and achieve sustainable management of forests [33]. In addition to impacting a main operation in forest management, timber harvesting, good or bad practices in forests, can also impact other parts of the ecosystem such as CO_2 sequestration, water cycles, local climates and the preservation of biodiversity.

E. Bell defines mathematical programming as "a generic term for a set of methods that can be used to optimise an objective in light of a set of constraints imposed on management activities or constraints imposed on the allocation of land to various uses" [34]. These exact techniques, including linear, goal and mixed integer programming, can be used to address spatial forest planning issues if computational needs do not become too demanding and are generally successful in solving simple planning problems [29].

According to Shan *et al.* [29], there are various forest planning techniques such as goal, integer, linear, mixed integer, non-linear, and dynamic programming. These exact methods aim to ensure optimal solutions and are typically used to validate heuristic methods or for small-sized problems. Considering all these techniques, the most common optimisation method for forest management planning is linear programming [35–37].

Linear programming is a useful tool to allocate resources efficiently, taking into account various restrictions and maximising or minimising a goal [38]. Kangas *et al.* [26] suggests that linear programming can be used to address a variety of sustainability issues, as it allows multiple constraints. This task requires two crucial elements: quantifying and ensuring the linearity of all variables needed to formulate the model.

Another traditional optimisation technique is dynamic programming [39]. This technique is more versatile and takes special account of the sequential decision-making process over time. As such, it becomes especially helpful to address and simplify the ongoing and difficult issues that require a long-term plan for forest management at any age [35, 40].

Exact conventional methods have proven their merit in optimising timber production and biodiversity conservation. However, while they may have the advantage of being transparent and interpretable, they may struggle to work for complex, non-linear relationships in forest stands, which is where more modern approaches based on heuristic methods can complement and enhance traditional forest management optimisation practices [41].

2.3 Heuristic Techniques

In the field of forest management, professionals are increasingly being tasked with balancing a broader range of resource objectives than ever before. Modern sustainable forest management has evolved from a focus on timber supply to incorporate more integrated land use planning, taking into account social, economic, and ecological considerations [42].

2.3.1 Multi-Objective Problem Solving

Despite significant advances in computing power and mathematical approaches, these traditional techniques are more difficult to adapt given the increasing diversity of forest planning objectives. The main challenge is the complexity of the problem, because the ecological and human components of decision-making have complex relationships [42]. Therefore, in response to computational deficiencies, techniques designed to generate good approximate solutions or predictions have been investigated [43]. These techniques may range from heuristic methods to newer approaches based on Artificial Intelligence.

In forest planning, pursuing optimal solutions is often a challenging and computationally demanding task. However, heuristics, defined as techniques that seek near-optimal solutions at minimal processing cost, provide a pragmatic solution to this problem, according to Reeves *et al.* [44].

Heuristics are particularly good at producing solutions that find a balance between computational economy and efficacy. Heuristics can also offer adaptability in problem formulation, allowing decision makers to align optimisation problems with diverse goals and preferences [45].

Categorised by solution generation strategies, heuristic methods can focus on point-based approaches such as Tabu Search and Simulated Annealing (SA), as well as population-based methods such as Genetic Algorithms and Particle Swarm Optimization (PSO) [29, 46].

In forestry, usually, the pursuit of multiple goals lacks a universally agreed criterion for measurement [37]. Assessing the degree of difficulty when meeting multiple goals in management plans remains a challenge. However, most forest management problems are cast as multi-objective optimisation problems. For this type of problem, multiple objectives need maximisation or minimisation, accompanied by meeting constraints to yield feasible solutions. The distinction between single- and multi-objective optimisation lies in the multidimensional space formed by objective functions alongside the typical decision variable space [37, 47]. When it comes to reaching the goal of the problem, heuristic optimisation techniques do not require a particular form for the objective function. Consequently, the design of the objective functions can vary from case to case.

According to Pukkala [48], there are four basic approaches to formulate the objective function for a multi-objective forest planning situation:

1. One goal is either minimised or maximised through the objective function, and the other goals become constraints.
2. The objective function measures the deviations of several objective variables from their target levels. These levels are given in other equations of the problem formulation.
3. A single objective is included in an objective function that is augmented with a penalty function.
4. A multi-attribute function is developed and used as the objective function.

Some of these formulations resemble conventional mathematical programming methods, yet they are solvable via heuristic optimisation techniques. Nonetheless, the planner commonly needs to define weights or constraints for the objectives. However, in this study [26], decision-makers do not establish these parameters in advance.

Given the inherent nature of heuristic problems, which often do not allow for exact solutions, researchers commonly evaluate the quality of the solution using statistical measures such as maximum value, minimum value, mean value, standard deviation and an estimated global optimum solution [29].

2.3.2 Genetic Algorithms

The term "Genetic Algorithm" was first used by John Holland in 1975 and laid the foundation for a thriving field of study that has since expanded to cover a wider range of applications than the initial GA [49]. The terms "evolutionary computing" and "evolutionary algorithms" are now used to refer to significant developments over the past ten years in the field, which have expanded the concept beyond the original GA [50].

Genetic Algorithms are population-based heuristic techniques based on natural selection and natural genetics. For this reason, the terminology of genetics and biology is used for their description [22]. This algorithm and its variations operate with a population of solutions, termed chromosomes, and combines them to generate new generations of solutions. It uses the concepts of selection, crossover, and mutation to simulate the mechanisms of biological evolution. These algorithms have been successfully used in a variety of areas, including music generation, genetic synthesis, strategic planning, and Machine Learning [51].

For forest planning, this algorithm provides flexible and efficient approaches to try to find optimal solutions. Given the fact that they search through large solution spaces efficiently, they are suitable for solving combinatorial optimisation problems. They can also handle both spatial and non-spatial objective variables, and so they are an efficient way to solve multi-objective optimisation problems [27].

The extensive use of Genetic Algorithms in forest planning is well documented in various studies and can be implemented in several ways [29]. However, a shared aspect of all implementations is their attempt to identify multiple solutions [52]. This means that GAs operate with a population of solutions. Each member of the population is a chromosome, representing a potential solution. Chromosomes can be viewed as linear formations consisting of contiguous segments, each segment corresponding to a variable within the problem. Each segment contains one or more elements, called genes, that perform a specific function [22]. From a biological point of view, each gene is responsible for expressing a specific characteristic associated with the individual chromosome.

The initial population of solutions is generated in a random manner and is subsequently subjected to simulated genetic evolution. The evolutionary process is carried out via three genetic operators: selection, crossover, and mutation.

Selection is an operation designed to give members of a population with greater fitness a greater chance of survival. In a numerical optimisation problem, the fitness of a solution is measured in relation to the objective function value. The selection process results in a population with an overall higher level of fitness. The population is then subjected to the crossover operation.

In this process, the genes of two parent chromosomes are combined to create a new chromosome. Finally, mutation involves the random alteration of one or more genes on one or more chromosomes in the population. In an optimisation context, mutation can help avoid regions of local optima [22].

Several studies have produced different results depending on the characteristics of the problem being solved. These characteristics include whether the problem was single-objective or multi-objective, how complex the constraints or objectives were, and whether the problem was integrated with spatial systems. Most of these studies compared the GA against other heuristic techniques, as previously described.

Bettinger *et al.* [53] and Pukkala *et al.* [54] conducted comparative studies evaluating various heuristic optimisation techniques in the context of forest management. They found that Genetic Algorithms are highly effective and their adaptability and robustness make them a valuable tool for managing forests, especially when dealing with complex spatial issues. Despite the fact that these studies were conducted, they did not investigate the effectiveness of these algorithms when dealing with uncertainty in forest planning.

In the study conducted by Ducheyne *et al.* [47], the objective was to implement an optimiser that facilitates the simultaneous optimisation of multiple objectives without the need to join these objectives in a single function.

In a multi-objective setting, an individual possesses multiple objective values, each pertaining to distinct objective functions. Consequently, the selection criterion for the evolutionary algorithm must be redefined to accommodate an objective value vector while retaining the essence of a single fitness value selection. The investigation compared two multiple objective algorithms: the Multiple Objective Genetic Algorithm (MOGA) developed by Fonseca *et al.* [55] and the Non-Dominated Sorting Algorithm II (NSGA-II) by Deb *et al.* [24].

The study [47] aimed to evaluate its performance against a random search strategy to gauge the efficacy of genetic operators. The findings revealed the effectiveness of both MOGA and NSGA-II in random search strategies. NSGA-II notably approached the Pareto-optimal front (solutions in which one of the objectives cannot be improved without sacrificing another one) more closely than MOGA.

However, MOGA excelled in discovering extreme solutions, contributing to a more widespread distribution along the Pareto front. Despite these strengths, neither algorithm managed to pinpoint extreme solutions. NSGA-II demonstrated a faster approximation of the front and displayed evenly spaced solutions, but MOGA slightly outperformed NSGA-II when considering the distance between extreme solutions. Nevertheless, NSGA-II exhibited greater stability in approximating the Pareto-optimal front.

Hypervolume assessments and additional analyses consistently demonstrated that NSGA-II surpassed MOGA in terms of solution space coverage. In general, NSGA-II performed more effectively in forest management challenges. The study concludes by highlighting the advantages of using Multi-objective Genetic Algorithms in forest management planning, and recommends integrating GIS (Geographic Information System) with heuristic methods to enhance efficiency.

In order to address the multiple objectives involved in land use management, a multi-objective evolutionary algorithm is the best solution. Multiple EAs, with slight variations in functionality, have been developed and implemented for these problems. The most widely accepted are MOGA [55], NPGA [56], NSGA, and NSGA-II [24].

In addition, Deb *et al.* [57] explored how NSGA-II can be used to optimise multiple, often conflicting, objectives within the context of land use management. NSGA-II was selected as the optimal solution and designated as NSGA-II-LUM (NSGA-II in Land Use Management). NSGA-II was chosen for its successful application to a broad range of problems.

The GIS-based spatial algorithm effectively evaluated three key objectives: maximising economic return, maximising carbon sequestration, and minimising soil erosion. These objectives are particularly important in addressing global warming and soil degradation, making NSGA-II-LUM a valuable asset in contemporary environmental research. Despite data limitations, NSGA-II-LUM showed potential to predict long-term global changes. It uncovers complex relationships and offers valuable information on land use management, promising for dynamic applications [57].

Fotakis *et al.* [22] introduced a spatial operator to enhance the efficiency of genetic algorithms in multi-objective spatial forest planning. The study used a constrained NSGA-II against a standard genetic algorithm to evaluate this newly proposed methodology. The focus was on a typical harvest scheduling problem, with the aim of maximising timber volume while minimising sediment levels, with constraints on minimum timber yield and even flows.

This algorithm adaptation included a spatial operator that functions as a secondary mutation mechanism with a low probability. The operator assessed the spatial relationships between the "mutated" land unit and its neighbouring units within the study area, ensuring localised spatial coherence in planning.

This modified algorithm, termed Spatial NSGA-II, achieved superior performance in both constrained and unconstrained optimisation problems. Not only did it generate better outcomes, but it also naturally preserved the cohesion of old forest patches, an emerging property that was not explicitly targeted as a goal.

According to Ducheyne *et al.* [58], when looking at the use of Genetic Algorithms to solve a multi-objective optimisation problem in forest management, it is also essential to take into account spatial objectives. Neglecting spatial intricacies during planning may lead to plans that are misleading or impractical. Genetic Algorithms, particularly when integrated with GIS, offer a solution by providing spatially feasible plans during the optimisation process itself.

Insights from Næsset and Kurttila [59, 60] further emphasise the importance of incorporating spatial objectives and constraints in optimisation of forest management.

The connection between the Genetic Algorithm and GIS through exchange files ensures platform-independent communication, aiding in optimal task allocation. The results of this type of integration revealed substantial differences in spatial layout and output levels when spatial information was included, although only a fraction of the Pareto front was explored [61]. This highlights the importance of incorporating spatial data within the optimisation process and demonstrates the potential of a GA – GIS fusion in the development of spatial decision support systems. It also underscores the risk of omitting spatial-dependent objectives during implementation, which could lead to misleading or unfeasible plans, even in a simple problem with two management compositions.

Previous research by the same authors demonstrated the efficiency of GAs over single-objective formulations in forest management optimisation. By integrating the spatial analysis capabilities of a GIS within the optimisation process, spatially feasible plans were constructed during the optimisation.

Various studies consistently emphasise that the performance of heuristic methods largely depends on the parameters that guide their search process. Identifying ideal parameter values is essential to achieve high-quality solutions [43].

2.4 Applications of Artificial Intelligence in Forestry

Artificial Intelligence (AI) is increasingly impacting forestry, particularly in areas like forest health monitoring, illegal logging detection, and biodiversity assessment. Machine Learning (ML), a key AI field, is being applied to improve data collection and analysis through remote sensing and image recognition, enhancing forest management practices.

However, in forestry, challenges such as limited data availability and complex ecosystems have slowed broader implementation. Nonetheless, ML is proving valuable in specific applications, such as studying tree volume and growth dynamics.

Some of the following studies show the recent advancements and applications in forest related tasks using Machine Learning models.

Decision tree-based models, such as Random Forest (RF) and Gradient Boosting Decision Trees (GBDT), have been widely used for biomass estimation and forest productivity modelling. For example, He *et al.* [62] applied RF to include more forest stand variables and environmental relationships, helping to improve prediction accuracy. For instance, in [63], the authors also utilised RF algorithms to forecast tree species distribution. This method has also been used for improving the accuracy of coarse resolution vegetation maps by down-scaling them to finer resolution climatic grids [64].

GBDT has been applied in forest resource management, where the integration of multi-temporal LiDAR data with climatic factors has been shown to enhance model performance for forest productivity, as demonstrated by Sotomayor *et al.* [31]. These algorithms showcase superior performance when compared to more traditional approaches.

Another important technique, Artificial Neural Networks (ANN), is used for tasks such as estimating tree volume, above-ground biomass, and evaluating the suitability of forest plantation sites. For example, Nitoslawski *et al.* [65] used an ANN to assess the suitability of cork oak plantations and predict biomass. Additionally, deep learning methods such as Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) have advanced forest monitoring. Peng *et al.* [66] highlighted their potential in processing satellite and drones data for soil quality assessment to help maximise biomass production while minimising environmental impact, and also facilitate crop monitoring.

Support Vector Machines (SVM) are also applied in forest management tasks such as tree stem taper mapping [65] and predicting the spatial evolution of forest land [67]. Furthermore, Decision Support Systems, which integrate various ML models, help optimise soil quality and predict environmental conditions to enhance forest productivity, as evidenced by Peng *et al.* [66].

ANN models have been widely used in fire risk prediction [68, 69], with studies such as those by Safi *et al.* [70] and Sakr *et al.* [71] demonstrating their effectiveness in modelling fire hazards and estimating fire spread dynamics. Other studies also reflect on the use of ANN to help model forest risks such as organic carbon stocks in the soil and evaluate snow damage [72].

Similarly, Support Vector Machines (SVM) have proven effective as an alternative to predict the level of fire hazard or to predict the days of spread of the fire, with studies by Liu *et al.* [41] and Safi *et al.* [70] showing its reliability in risk modelling. However, all of these studies explain that challenges remain due to data availability, interpretability concerns, and the need for widespread adoption.

A more advanced model, Particle Swarm Optimization Random Forest (PSO-RF), has also been applied to fire risk prediction.

Integrated with RF, PSO-RF combines environmental variables such as altitude, slope, and proximity to human activity to improve prediction accuracy, as demonstrated by Shi *et al.* [73]. The integration of the PSO algorithm with the RF model demonstrates the effectiveness of combining optimisation algorithms with traditional models. The study also compares other Machine Learning models that include Logistic Regression, Support Vector Machine, and Random Forest. However, the PSO-RF model outperforms the others, as evidenced by higher Area Under the Curve (AUC) values.

Furthermore, according to Schmoldt *et al.* [74], several Machine Learning algorithms such as Knowledge-Based Systems (KBS), Artificial Neural Networks, Fuzzy Logic and Bayesian Belief Networks (BBN), can be used to predict fire behaviour, assess the probability of pest and disease outbreaks, forecast weather events, and assess the impact of anthropogenic activities, all of which affect optimal management of forests.

2.5 Summary

This chapter analysed forest resource management methods, focussing on the evolution from traditional optimisation techniques to AI-based and heuristic approaches. It highlights the challenges of multi-objective planning, spatial dependencies, and computational complexity in forest management.

The shift from linear and dynamic programming methods to heuristic techniques demonstrates a continuous effort to improve decision-making. Recent advances in Artificial Intelligence and Machine Learning further demonstrate significant potential in this field.

The analysis in Table 2.1 reveals that traditional methods, such as linear and dynamic programming, although effective for smaller or simpler problems, encounter limitations when applied to more complex or dynamic forestry scenarios. Heuristic approaches, including Genetic Algorithms and NSGA-II, have emerged as strong alternatives capable of addressing multi-objective and spatially dependent forest planning challenges. However, these methods require careful parameter tuning to ensure optimal performance. Artificial intelligence and Machine Learning methods have shown significant promise in predictive modelling and monitoring tasks. However, their use for optimizing resource management in forests is not widely documented.

As a result, it can be concluded that there is a current knowledge gap concerning systems designed to create forest management alternatives that are both easily accessible and open to modifications.

Table 2.1: Summary of methodologies applied in forest scenarios

Method	Strengths	Weaknesses
Linear Programming	Interpretable, suitable for small-sized problems	Cannot handle complex, dynamic, or spatial dependencies
Dynamic Programming	Suitable for sequential decision-making over time	Computationally intensive for large-scale or long-term plans
Genetic Algorithms	Flexible, handles multi-objective and spatial optimisation	Sensitive to parameter settings, requires significant tuning
NSGA-II	Effective for multi-objective problems, generates diverse Pareto-optimal solutions	Computational complexity in large-scale applications
AI/ML Models	Highly accurate predictions, helpfull in risk assement	Limited by data quality and availability, scalability remains a concern, not usually used for optimisation tasks

3 TOOLS AND FRAMEWORKS

The successful implementation of this project is based on the use of contemporary tools and frameworks that facilitate the flow of development processes, improve operational efficiency, and support the execution of complex processes. The following sections provide an overview of the main technologies employed, including development environments, web frameworks, and specialised libraries that are essential for the implementation of front-end and back-end functionalities.

3.1 Web Platform Details

This research is part of the F4F project [`https://f4f.serq.pt`], within Pilot Projects 7 and 10. Previously, it was proposed to create a web platform that combines the work of other projects, including this simulator. This platform, named *Floresta Digital*, is the result of projects by some ISEC students: Daniel Correia, José Lopes and André Guimarães [75]. Ana Rita Malta was responsible for the creation and development of the initial platform. This project also counts on the participation of Professor Mateus Mendes, Professor Raúl Salas González, and Professor Beatriz Fidalgo.

The main goal of the web platform was to integrate a suite of simulators tailored to the forestry domain, with the aim of making them accessible to a diverse range of potential users, including forest managers, foresters, forest owners, and students.

The platform hosted three different types of simulator, each catering to specific aspects of forest resources. The first is a growth and production simulator that runs on the maritime pine stand. The second is a tree-level simulator that estimates the volume of cork produced by cork oak, and there is also a similar tree-level simulator that predicts the volume produced by maritime pine. There is a fourth simulator specifically designed for the estimation of American oak volume that is still under construction.

Through the efforts of this project, the website now has a total of four functional simulators, classified into two main categories: volume or area simulators and growth or management alternative simulators.

3.2 Development Environment

The choice of an appropriate development environment was critical to enable efficient coding, debugging, and collaboration. PyCharm [76] (version 2024.3) was used as the primary Integrated Development Environment (IDE) due to its advanced features, such as intelligent code assistance, integrated debugging tools, and seamless integration with version control systems. Its robust support for Django further reinforced its suitability for this project.

Version control was managed through GitHub [77], which served as a centralised repository to track changes and maintain code integrity. Although the development process was conducted individually, the repository was structured to facilitate future collaboration. A GitHub Action was implemented to automate server-side updates, ensuring that the latest version of the project is deployed efficiently and consistently upon pushing changes to the main branch. To manage tasks and monitor progress, Trello [78] was used as a project management tool, allowing clear organisation of workflows and effective timeline management.

3.2.1 Back-End Framework and Deployment Tools

The back-end development of the project was centred around Django (version 4.2.9) [79]. Launched in 2003, Django is a high-level Python web framework known for its scalability, security, and extensive built-in features [80]. Supports swift development and the design of applications using clear and functional design principles. The framework architecture provides an integrated Object-Relational Mapping (ORM) system, an intuitive administrator interface, and middleware capabilities for efficient handling of requests and responses [80]. In particular, Django is freely available and adheres to open source principles, fostering community collaboration and continuous improvement [81].

The deployment process used Nginx (version 1.26.2) [82] and Gunicorn (v23.0.0) [83] to establish a high-performance, secure, scalable production environment. Nginx was configured as a reverse proxy server, facilitating efficient load balancing, secure handling of client requests, and the serving of static files. Gunicorn was used as the Web Server Gateway Interface (WSGI) HTTP server. This tool communicates between the Django framework and Nginx to ensure a reliable handling of HTTP requests, illustrated in Figure 3.1.

To streamline the deployment workflow, a GitHub Action was implemented, automating the process of updating the codebase on the server. This integration eliminated the need for manual interventions outside the local development environment, as well as automatically linting the codebase, ensuring consistency and reducing deployment overhead.

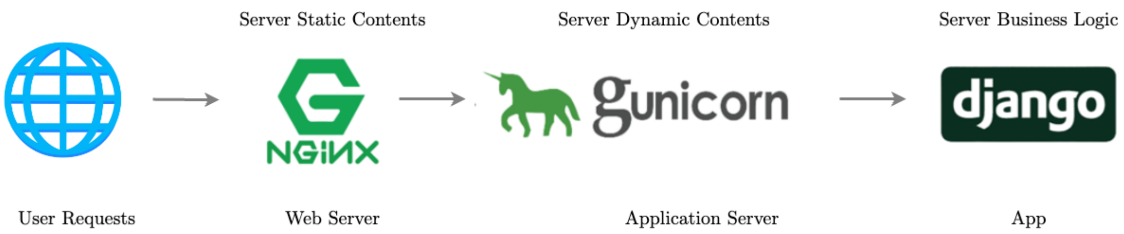


Figure 3.1: Interactions of a Django app with Gunicorn and Nginx

3.2.2 Front-End Integration

The front-end aspects of the project were developed using the Django templating engine, which provided a solid foundation to render dynamic content. To further enhance interactivity and user experience, Django-Leaflet (version 0.29.0) was integrated to implement interactive mapping capabilities, allowing intuitive visualisation and manipulation of spatial data.

For design and layout, Bootstrap (version 5.2) and FontAwesome (version 5.0) were used to ensure a responsive, consistent, and visually appealing user interface. These frameworks streamlined the development process by offering pre-designed components and iconography. Furthermore, GSAP (v3.12.5), a widely adopted JavaScript animation library, was utilised to introduce engaging animations while maintaining compatibility with all major web browsers [84].

Other Django tools employed include Django's built-in static file management system to efficiently serve front-end assets and Django's Form and Widget framework, which facilitated the creation and customisation of user-friendly input forms.

3.2.3 Python Libraries

A diverse range of Python libraries were integrated into the project to support various computational, analytical, and visualisation tasks. For data manipulation and analysis, Pandas (v2.2.0) was used due to its robust handling of structured data. Visualisation tasks were performed using Matplotlib (v3.8.2) and Seaborn (v0.13.2), which allowed the creation of informative graphical representations. Additionally, Folium (v0.15.1) was used to produce interactive geographic maps.

Arithmetic operations were supported by libraries such as NumPy (v1.26.3) and SciPy (v1.14.0). Optimisation tasks, including those that involve genetic algorithms and multi-objective optimisation, were handled using Pymoo (v0.6.1.1) [85].

4 ARCHITECTURE OF THE WEB PLATFORM

This chapter presents the redesign process of the existing website, which required a comprehensive restructuring to improve functionality, resolve persistent bugs, and enhance the user experience. The structure of the previous website, although functional, was constrained by limitations that affected performance and scalability. The following sections outline the newly developed code structure, detailing the key changes that address the aforementioned issues, as well as the primary defects that were identified and resolved during the redesign process. Furthermore, new wireframes are presented to illustrate the updated layout and improved navigational flow, which reflect the project's focus on creating a more intuitive, user-centred interface.

4.1 Principles for Efficient Web Development

Creating a successful website requires following a number of best practices, each designed to improve its functionality, usability, and overall performance [86]. First, it is essential to establish clear objectives that outline the purpose, target audience, and desired results of the website. It is a fundamental step in all subsequent design and development decisions. User-centred design principles are key, guiding the creation of intuitive navigation structures, clear layouts, and accessibility features that focus on user experience [86]. Ensuring responsiveness across devices and screen sizes is essential, emphasising the importance of having a mobile-friendly design in the current digital environment.

Implementing SEO strategies is vital for maximising the website's visibility and organic traffic. This requires thorough keyword research, meta tag optimisation, and the creation of relevant high-quality content that resonates with both users and search engine algorithms [87].

Beyond user experience and search engine optimisation, website performance is a critical consideration. Employing techniques to minimise loading times, such as file compression, browser caching, and content delivery networks, is instrumental in fostering a seamless browsing experience.

Another critical component of modern web development is the implementation of a Continuous Integration and Continuous Deployment (CI/CD) pipeline. CI/CD is a development practice that automates the processes of integrating code changes, testing, and deploying them to production environments [88].

This approach is indispensable in ensuring that updates are seamlessly deployed without compromising website stability or user experience. By continuously validating and deploying code changes, developers can address issues early, reduce downtime, and maintain a reliable, scalable platform. CI/CD is not just a convenience, but a necessity in modern web development, allowing teams to work efficiently and consistently deliver high quality software [88].

Security measures cannot be overlooked, with encryption, regular software updates, and secure authentication mechanisms serving as foundational pillars to protect against cyber threats. Equally crucial is the ongoing maintenance and updating of the website's content, software, and security features to ensure its continued relevance, functionality, and security.

Upon reviewing the platform and scrutinising its underlying source code, it became apparent that a portion of the project should be allocated to enhancing both its front-end interface and the structural integrity of its codebase. Consequently, the web development portion of this project was divided into two distinct initiatives: first, the enhancement of code authored by other programmers, focussing on optimisation and organisation, and secondly, the development of a simulator dedicated to forest management alternatives.

4.2 Updated Website Infrastructure

The previous website was a collection of projects developed by different individuals, including bachelor and master students, brought together on a single platform. As developers worked independently and were not part of a cohesive team, the integration process posed some challenges. As a result, the application was not fully aligned with standard practices, as previously noted.

To address these issues, three main tasks were undertaken. First, a version control system was implemented to maintain a history of changes, as previous work had been done locally without proper tracking. Second, a new project was initiated with a clean structure that followed the framework guidelines and best practices for web development. Finally, a CI/CD pipeline was implemented using GitHub Actions to automate testing and deployment, ensuring the codebase was correctly validated and updates were seamlessly deployed to the server while keeping the website operational.

The original project was found to have some areas for improvement in responsiveness and visual appeal, along with a few issues identified during testing. As a result, the goal of the final task was to redesign the website, enhancing its aesthetics and addressing the identified issues.

4.2.1 Implementation of Version Control

Incorporating version control into a website project is essential to facilitate collaboration, track changes, organise code, manage risk, enable experimentation, and ensure the quality of the code.

To achieve this, Git was chosen as the version control system. Git is a distributed version control system that tracks changes to any set of computer files. It is typically used to coordinate the work of multiple developers who work together to manage and develop software [89]. Git allows for flexible branching, enabling developers to work on new features or fixes in isolation before merging them back into the main project.

The project is hosted on GitHub, which serves as a central cloud-based repository. GitHub provides an online platform for storing code, managing collaboration, and tracking changes through commits, branches, and pull requests. It allows multiple developers to contribute by pushing their changes and creating pull requests, which can be reviewed before merging into the main codebase. GitHub also offers additional features such as issue tracking, project management, and integration with CI/CD tools for automated testing and deployment.

On the server, a private Git repository is set up, which is linked to the GitHub repository. This allows the server to pull the latest changes from GitHub or a specific branch and deploy them directly. The GitHub repository is used for deployment purposes, allowing the application to automatically pull the updated code and push any necessary changes back to the the server. This setup ensures that the local copy of the project from the server is always up-to-date with the central repository, maintaining consistency between environments.

Using Git both locally on the server and remotely on GitHub, the project is now set up for seamless collaboration, efficient code management, and easier tracking of changes. New developers can quickly get started, maintain a detailed history of the project, and contribute to the codebase while reducing the risk of code loss. This setup also provides a secure and organised way to manage the project development lifecycle.

In addition to implementing version control, a Continuous Integration and Continuous Deployment (CI/CD) pipeline was set up to automate the process of testing, reviewing, and deploying code changes. This pipeline ensures that every commit is properly reviewed, validated, and deployed to the server with minimal manual intervention, reducing the potential for errors and increasing the speed and reliability of the development process.

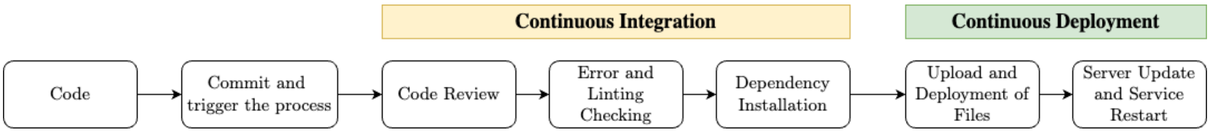


Figure 4.1: CI/CD pipeline in the project

The pipeline operates in several stages, each with a specific purpose, as shown in Figure 4.1.

The pipeline begins with Code Review and Approval. Every time a commit is made to the repository, the pipeline triggers an automatic code review process. This step involves developers reviewing the code to ensure that all changes adhere to the project's coding standards and best practices. Only after the changes are approved does the pipeline proceed to the next stages. This process guarantees that only high-quality, reviewed code is integrated into the main codebase.

The next stage is Python error check and linting, where the pipeline automatically scans the code for errors or potential issues. This is a critical step in maintaining the quality of the code throughout the project. Python linters are used to identify coding errors, inconsistencies, and formatting problems. In addition to linting, tests are executed to verify that changes do not introduce bugs or regressions. If errors are found at this stage, the pipeline stops and flags the issues for review. Only when all errors are resolved can the pipeline move forward.

Once the code passes the error check step, the pipeline performs Dependency Installation. Using a *requirements.txt* file, the pipeline ensures that all required Python libraries and packages are installed in the project's environment. This step guarantees consistency between the local development setup and the deployment environment, ensuring that the application has all the necessary resources to run properly on the server.

After the dependencies are installed, the pipeline moves to the Upload and Deployment of Files stage. During this phase, the updated files are securely uploaded to the server. This ensures that only the necessary files are transferred, minimising disruptions and keeping the deployment process efficient.

Finally, the pipeline executes the Server Update and Service Restart stage. After the files are uploaded, the server environment is updated by restarting the relevant services. This typically involves restarting the application server, such as Gunicorn, and reloading the web server, such as Nginx, to reflect the new changes. These updates ensure that the changes deployed are immediately live and accessible without any downtime or disruption to the website's availability. If any issues are encountered during the deployment, the process is stopped, and the problem is flagged for review to ensure the integrity of the deployment process.

By implementing this CI/CD pipeline, the development process has been significantly streamlined. The code is thoroughly reviewed, tested, and automatically deployed with each commit. This reduces the chances of errors that slip through cracks and accelerates the time from development to production while maintaining a high level of code quality and server stability. The automated nature of the pipeline also reduces the need for manual intervention, allowing developers to focus on building new features and improving the application without worrying about deployment-related issues.

4.2.2 Project and Website Organisation

In Django, the organisation of a website with multiple applications involves structuring the project in a way that promotes modularity, scalability, and maintainability. This approach allows for better code organisation and easier management by ensuring that each app handles a specific functionality within the project.

The project was previously organised as described in Table 4.1. While functional, the structure could be improved, as some folders included a mix of temporary, environment, and personal files, which obstructed navigation and maintenance.

The project is now divided as Figure 4.2 describes.

The Accounts folder contains everything related to user registration and login. It also contains email functionality. The Setup folder also contains global project settings. For security reasons, it is good practice to keep the settings for the development and production environments separate. This folder is also the location for the definition of web page URLs.

The Media folder is for the storage of files uploaded by users. For example, some simulators may ask users to upload a picture of a tree.

In the case of the main project folder, there are several key components. Firstly, the Migrations folder manages database schema changes using Django's migration system. Second, the Templates folder stores HTML files that make up what the user sees when accessing a web page. Django's template inheritance allows common layout elements such as navbars and footers to be shared across templates.

Table 4.1: Previous organization of the Django project for Floresta Digital

Folder Name	**Description**
Fichas	Contains general data sheets for the project
Ficheiros	Stores files for the "Rede de Povoamentos" page
Manual	Contains miscellaneous files
Temp	Temporary files
account	Handles user account-related functionality
blog	Stores files and code for a deleted blog section
calculadora	Code and resources for a simulator
mapa	Generates and displays a map
media	Stores uploaded media files
media_cdn	Serves static copies of media files
myenv	Configuration and environment files for the app
mysite	Contains main project settings and configuration files
newenv	Additional environment configuration files
personal	Miscellaneous or personal files
simuladorCarvalho	Code and files for the "Carvalho" simulator
simuladorPB	Code and files for the "Pinheiro Bravo" simulator
srcenv	Environment files specific to the source code
static	Stores static files such as images, CSS, and JavaScript
templates	Contains reusable HTML template files
volume	Code and resources for the cork volume simulator

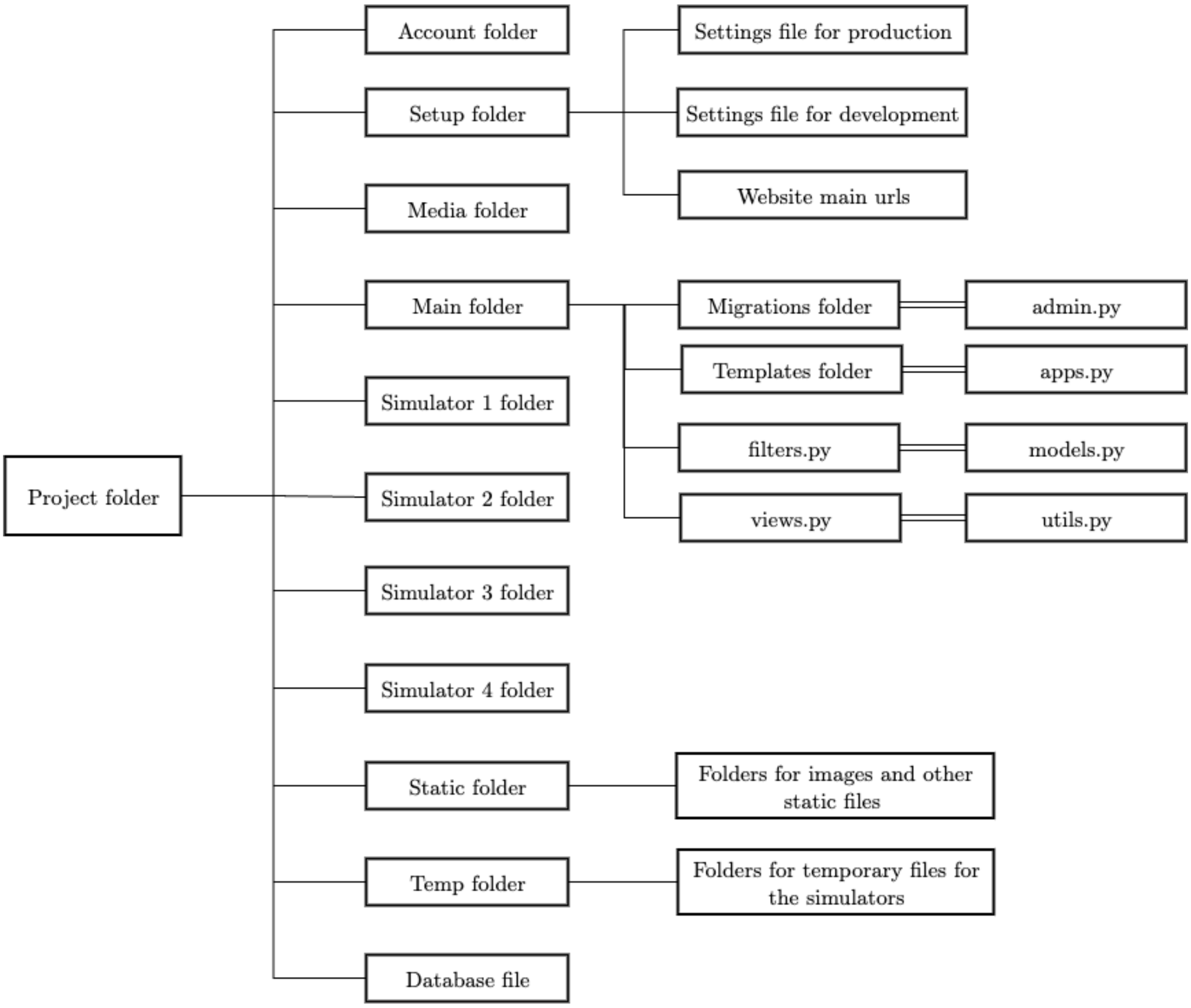


Figure 4.2: Current organisation of the project

In addition, the Filters file contains custom filters used for data manipulation or formatting in Django templates. The Views file contains view functions or classes. These are responsible for handling HTTP requests and rendering responses. The Admin file contains configurations for registering models with the Django admin interface, enabling CRUD (Create, Read, Update, Delete) operations.

The App file also contains configurations for the Django application, including settings, URLs, and application-specific middleware. The Models file defines the schema and data structure for the Django application. While the Utils file is not automatically created, it is commonly used to house functions for use in different contexts.

The Simulators folders are typically called apps in Django applications. While the main pages are stored in the "Main" folder (Main application), each simulator is treated as a single application, inheriting the main template but with a different website body. The structure of these folders is based on the main folder, but includes additional files for the specific needs of each simulator.

Additionally, the Static folder is dedicated to storing static files (CSS, JavaScript, images, etc.) required for the website, which are not expected to change in production. The Temporary folder is used by simulators to store files temporarily before saving the data to the database. The files in this folder should be deleted after they have served their purpose. Lastly, the Database file contains the details of the MySQL database.

Structure of the Website

A typical website consists of several linked pages designed to make the website easy to use and to interact with. At its core, the home page acts as a "front door", providing an initial overview of the website's purpose, content, and navigation options. The website "Floresta Digital" features four primary pages accessible through the website navbar: "Sobre" (About), "Rede de Povoamentos" (Stands Grid), "Simuladores" (Simulators), and an account management page, as illustrated in Figure 4.3.

Additionally, users can access the homepage, which serves as the main entry point, offering an initial overview of the website's purpose and navigation options. The "Sobre" page provides detailed information about the project and associated research. Under the "Rede de Povoamentos" sub-menu, users find "Ficha Geral de Povoamentos" (General Stands Sheet) and "Mapa Geral de Povoamentos" (General Stands Map), which explain forest stand concepts and provide information in PDF format. The "Simuladores" page offers an overview of available simulators, categorised by their goals. Access to the simulators is restricted to logged-in users as a security measure.

Finally, the Account page, or the Private Area, behaves differently based on user status.

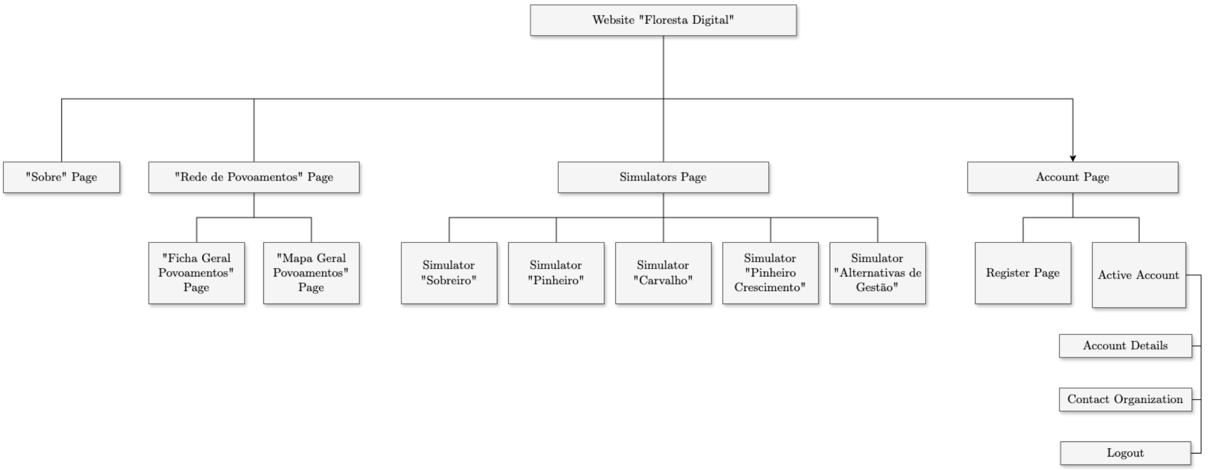


Figure 4.3: Published website pages

If not logged in, the link redirects to the Login/Register form, while logged-in users can view their username in the menu link and access options such as account details modification, organisation contact via a form, database access (if permitted) and logout functionality.

4.3 Improvements to the Existing Codebase

Substantial improvements were implemented in the existing codebase to address critical deficiencies and increase its functionality, security, and maintainability. These updates have resulted in a more robust and efficient system, tailored to meet the needs of users and ensure long-term reliability.

A key focus was improving security measures for sensitive information, such as secret keys. These were relocated from source code to environment variables, safeguarding them from exposure during code sharing or deployment in production environments. This change aligns with best practices for secure software development.

The file management system underwent significant restructuring to ensure scalability and user isolation. The previous implementation lacked organisation, making it difficult to maintain a clear structure for user-generated files. The updated mechanism now associates files with individual users, allowing multiple users to run simulations concurrently without interference. This ensures that files are securely processed and accessible only to their respective users. Future improvements could involve migrating temporary files to a database, enabling the creation of a comprehensive user history.

To optimise resource usage, a system for automatic deletion of unused temporary files was introduced. Temporary files generated during simulations or other operations are now periodically removed via a scheduled clean-up function.

This prevents unnecessary consumption of server storage and maintains an organised directory structure.

For better monitoring and debugging of the system, a logging mechanism was integrated. These logs provide a record of errors and critical events, offering valuable insight for troubleshooting and system performance analysis. Additionally, this change resolved the issue that errors were printed directly onto the server console, which previously caused crashes.

The database structure was thoroughly reviewed and refined. Data redundant and outdated were removed and the model rules were adjusted to ensure consistency and integrity. These updates improved the reliability and alignment of the database with the evolving requirements of the project.

Improvements were also made to account management functions, simplifying their structure and eliminating inefficiencies. These changes enhance the system's intuitiveness, reduce potential bugs, and provide a smoother user experience.

The user contact forms were revised to ensure reliable communication between users and the organisation. Their functionality and usability were tested extensively, addressing previous issues and improving their responsiveness.

Finally, comprehensive testing and refinement of existing simulators were conducted. Layout issues and functional errors were identified and resolved, ensuring that the simulators operated correctly.

Together, these improvements ensure that the platform can meet growing demands while maintaining high performance, reliability, and ease of use.

4.4 Enhancing User Experience

In order to improve the user experience on several fronts, the website optimisation initiative was carried out methodically. First, a reduction in the JavaScript and CSS files was implemented to streamline the code complexity. This resulted in better page load times and simplified maintenance. This reduction not only improved the website's performance but also allowed for smoother navigation.

The responsive design principles were then applied to ensure seamless adaptability to different screen sizes and devices, especially for mobile users. This optimisation strategy boosted accessibility and usability across platforms. To increase visual engagement and aesthetic appeal while maintaining design consistency across the website, content enrichment through the addition of images was prioritised.

In addition, minimalist design principles were adopted to prioritise simplicity and clarity. Ultimately, this enhanced content readability and user comprehension.

These refinements were followed by extensive usability testing to ensure compliance with accessibility standards and provide an inclusive browsing experience for all users.

In general, coordinated optimisation efforts improved readability, accessibility, and usability, resulting in a better browsing experience across demographics and devices.

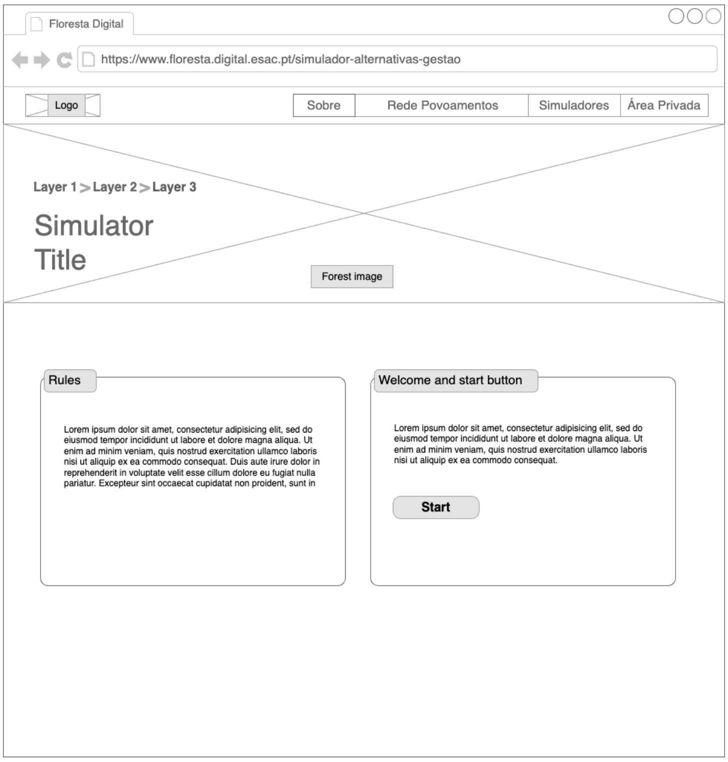
Before making the changes mentioned previously, it was necessary to create wireframes for the website in general and in particular for the simulator.

Mockups or wireframes serve as visual blueprints, illustrating the layout, functionality, and user flow of the webpages prior to coding. They offer several benefits. First, they facilitate early feedback from stakeholders, allowing users and developers to identify potential usability issues and suggest improvements before the start of significant development work. They also help to ensure that the user interface is consistent with the overall design vision and follows best practices for clarity and ease of use. Ultimately, mockups foster clear communication between designers and developers, creating a shared understanding of the intended functionality and user experience before diving into technical implementation [90].

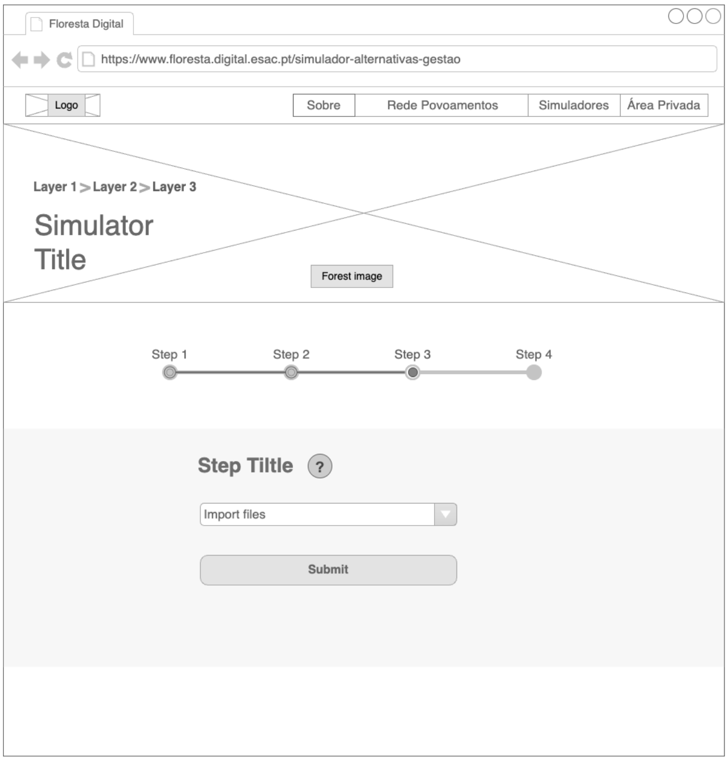
The use of wireframes was critical to ensure consistency throughout the design of the website, giving it a polished appearance. In addition, these mockups were useful in understanding the mandatory steps of the form, which are required for subsequent processing and presentation of results. They helped create a user-friendly interface that was consistent with the overall goals of the simulator. In the end, the development of a clear vision prior to execution prevented any technical conflicts, thus optimising the coding process.

Figure 4.4 presents the wireframes created for this simulator, designed to align with the style and functionality of other simulators on the platform. Additional wireframes are provided in Appendix A. The image in 4.4a shows the mock-up of the welcome page common to all simulators. Upon opening a simulator, users are greeted with this welcome page, which provides an explanation of the rules and, when necessary, offers example files (displayed in the left card). The right card or box typically contains the start/next button, guiding users to the subsequent form where they can input data.

The wireframe shown in 4.4b represents the basic structure of the form. First, it was essential to integrate a navigation component in the form of a step bar into the web page. This feature allows the user to move seamlessly between steps while remaining aware of the phases required to complete the simulation. Beneath the step bar is a dedicated area that contains the form fields. Although the inputs may vary from step to step, they all follow a consistent format: each step is accompanied by a title and an adjacent help button with hidden tips. After filling in the form, the user is presented with a submit button, signalling the end of the input process.

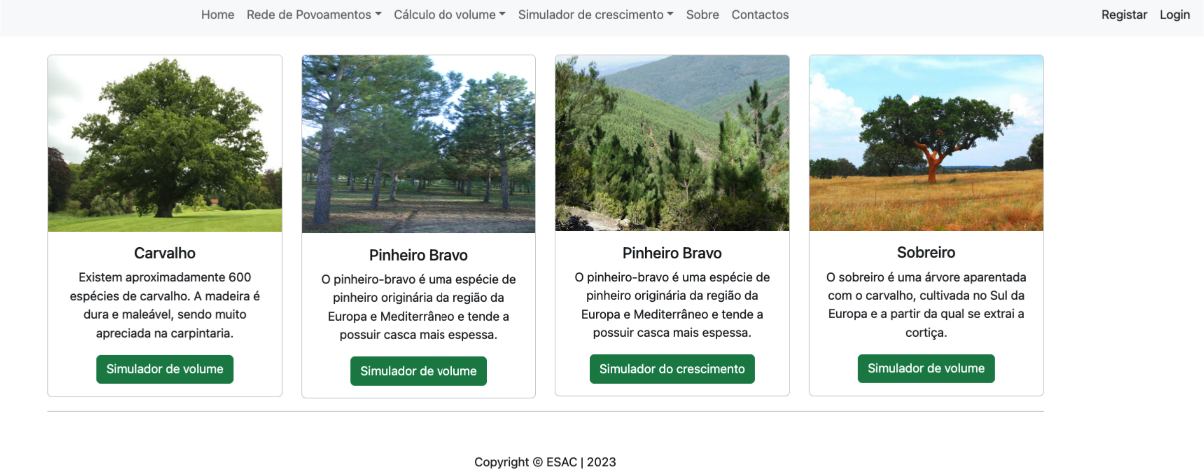


(a) Mock-up of the simulator welcoming page



(b) Mock-up for submitting files during several steps of the form

Figure 4.4: Example wireframes for the simulator



Carvalho

Existem aproximadamente 600 espécies de carvalho. A madeira é dura e maleável, sendo muito apreciada na carpintaria.

Pinheiro Bravo

O pinheiro-bravo é uma espécie de pinheiro originária da região da Europa e Mediterrâneo e tende a possuir casca mais espessa.

Pinheiro Bravo

O pinheiro-bravo é uma espécie de pinheiro originária da região da Europa e Mediterrâneo e tende a possuir casca mais espessa.

Sobreiro

O sobreiro é uma árvore aparentada com o carvalho, cultivada no Sul da Europa e a partir da qual se extrai a cortiça.

Copyright © ESAC | 2023

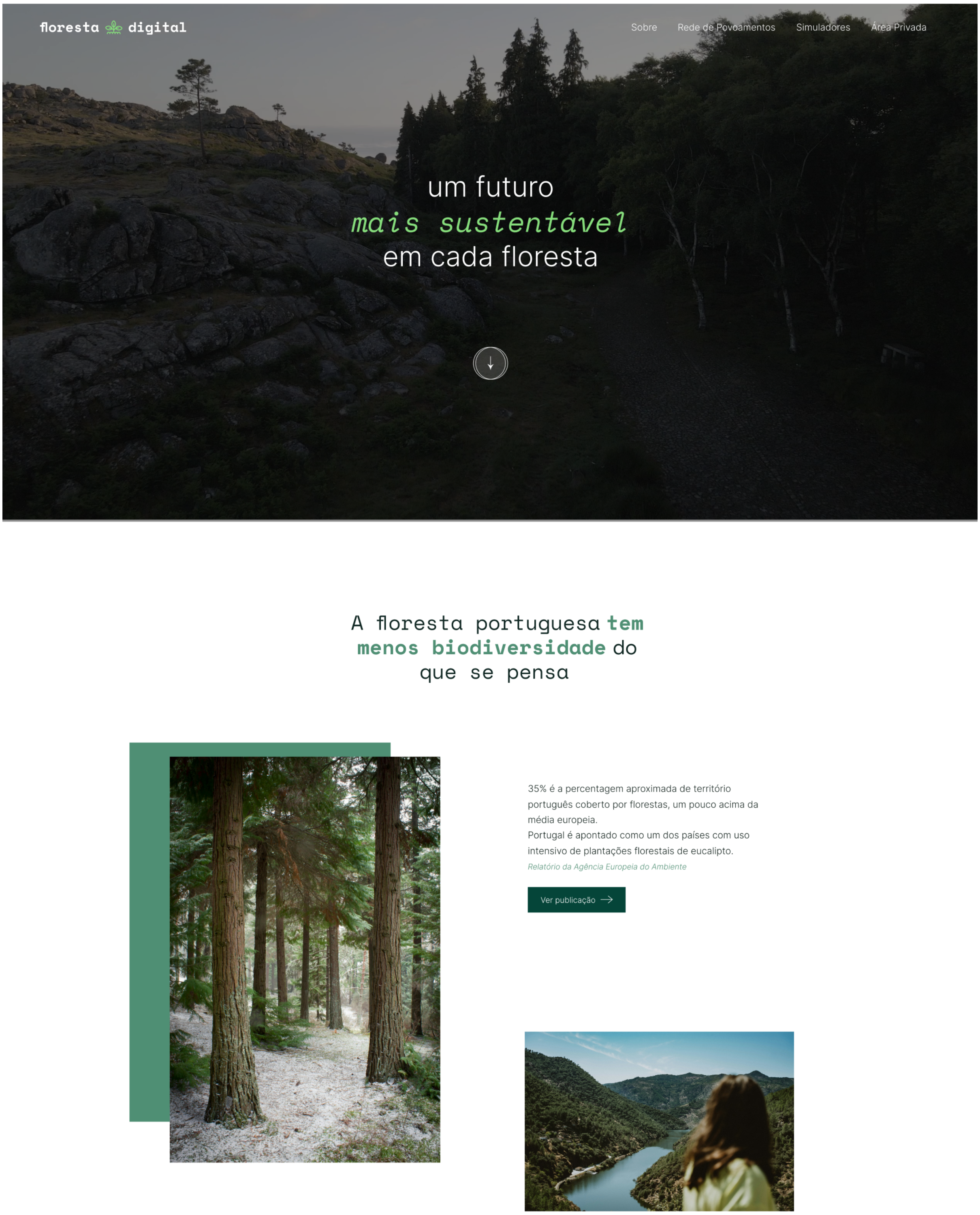
Figure 4.5: Website "Floresta Digital" previous homepage[1]

After submission and subsequent validation of the data, the users are redirected to a page where they are greeted with messages indicating success or error, as shown in Figures 5 — 6. In the absence of errors, users can proceed to the next simulation step, repeating this process until the completion of the form is reached and the results are presented.

Finally, Figure 4.6 provides a snapshot of the current homepage design of the website. For a more detailed view, visit [`www.floresta.digital.esac.pt`]. In contrast, Figure 4.5 presents the layout of the previous homepage of the website, highlighting the differences in design and structure.

---

[1] Accessed in January 2024 at `https://www.floresta.digital.esac.pt/`, with content changed since then



um futuro
mais sustentável
em cada floresta

A floresta portuguesa **tem menos biodiversidade** do que se pensa

35% é a percentagem aproximada de território português coberto por florestas, um pouco acima da média europeia.
Portugal é apontado como um dos países com uso intensivo de plantações florestais de eucalipto.
Relatório da Agência Europeia do Ambiente

Figure 4.6: Sample of the new "Floresta Digital" homepage

4.5 Summary

This chapter outlines the redesign of the web platform, focussing on resolving prior limitations, improving usability, and improving system efficiency. Key updates include implementing version control using GitHub, introducing a CI/CD pipeline for automated testing and deployment, and restructuring the project for better scalability and maintenance.

The platform codebase was refined to improve security and optimise file management. The simulators and user account management functions were improved to ensure reliability and user isolation. Usability improvements included adopting responsive design, minimising unnecessary files, and improving the visual layout. Wireframes were developed to guide the design process and ensure consistency across the platform.

These initiatives led to the creation of a modern and user-friendly platform that adheres to current web development standards and addresses various user needs.

5 DEVELOPMENT OF A SIMULATION TOOL FOR FOREST MANAGEMENT

This chapter outlines the design and development of the simulator inserted into the "Floresta Digital" project. The simulator automates previously manual simulations for efficient forest data management, enabling users to input detailed forest management options. The chapter is organised into sections detailing the system's core components.

In Section 5.1, the requirements that guided the simulator's development are outlined. Section 5.2 focusses on the database architecture, explaining how the data is structured and stored to support the functionality of the simulator. Finally, Section 5.3 describes how the simulator guides users through its multistep process and integrates user input with back-end processes.

5.1 Simulator Requirements

A clearly specified set of requirements is essential for the successful creation of any web tool. This section outlines the key requirements that guide the development of the simulator tool. Functional requirements define the essential tasks the system must perform while non-functional requirements, on the other hand, focus on ensuring the simulator is reliable and secure.

5.1.1 Functional Requirements

In the development of any simulator or form, functional requirements serve as a foundation, outlining the essential capabilities and tasks that the system must perform to achieve its objectives. These requirements are pivotal in ensuring that the simulator/form operates effectively, addressing user needs, and achieving the intended purpose.

Outlined below are the functional requirements for the simulator, presented in a structured format:

1. Access to the simulator shall be restricted to users with valid accounts on the website.
2. The simulation results must be presented in a clear and structured format, utilizing visual elements such as graphs, tables, and text summaries to enhance the user's understanding.

3. The simulator shall include a dedicated section containing example files and comprehensive instructions to guide users effectively.
4. The simulator shall have an entry page that includes:
    (a) A description of the rules of use.
    (b) A downloadable starter example file.
5. The simulator's workflow shall be logical and intuitive, segmented into distinct phases:
    (a) Import Forest Stands
    (b) Choose Stands to Simulate
    (c) Add Inventory Data
    (d) Define Planning Factors
    (e) Results Presentation
6. All user inputs must be validated to ensure compliance with defined formats, allowed ranges, and constraints specified by the underlying model.
7. The simulator shall handle invalid input by providing precise and actionable error messages, describing the issue, and guiding users to corrections.
8. User progress must be saved and preserved, unless they explicitly choose to initiate a new simulation.
9. The simulator must support simultaneous usage by multiple users, even under the same account credentials.
10. Dynamic elements are displayed or hidden on the basis of user input or computed results, ensuring a tailored and relevant user experience.
11. The simulator must provide functionalities for generating reports and downloading results in accessible formats, such as PDFs or CSV files.
12. A success or error message shall be displayed for each operation, confirming or clarifying the outcome.
13. The simulator shall display the form steps at all times to inform the user of their position in the workflow, while preventing navigation to blocked steps.
14. Each step in the simulator shall include a dedicated information section explaining its purpose and requirements.
15. The simulator shall display information about imported forest stands in a table and on a map, allowing users to select or deselect stands using either method. At least one stand must be selected to proceed to the next step.

16. In the inventory data step:
    (a) The user must choose between importing a file or manually adding data.
    (b) For manual entry, only the chosen forest stands will appear in the input form.
    (c) Inputs for manual inventory data must adhere to business logic constraints, such as limiting available species based on stand composition.
    (d) After adding data, the simulator must validate its accuracy.
    (e) For errors detected during file imports, the simulator should display warnings and offer the form-based option to correct errors.
17. The planning factors are defined as follows:
    (a) If a stand contains species with a current age of 0, users shall be able to add up to three alternative species compositions.
    (b) The user must define the range within which all cutting alternatives for the forest stand will be created.
18. The results shall be presented as follows.
    (a) Interactive visualisations such as plots.
    (b) Text summaries.
    (c) An option to download a complete report containing all simulation details.

5.1.2 Non-Functional Requirements

Non-functional requirements specify the qualitative attributes of a system, concentrating on its performance characteristics rather than its functions. These requirements guarantee that the simulator functions with efficiency, reliability, and security, while offering a smooth user experience. The non-functional requirements for the simulator are listed below:

1. The visualisation of simulation outcomes should be optimised for performance to ensure responsiveness when rendering large datasets.
2. The simulator interface must maintain a consistent design across all pages, using a clean layout and standard UI elements for ease of use.
3. Navigation between workflow steps should occur without noticeable delays, with feedback provided for actions requiring longer processing times.
4. Input validation must occur in real-time where feasible to provide immediate feedback and reduce user error.
5. Progress saving functionality must ensure data integrity and prevent loss of user data due to unexpected interruptions.

6. Simultaneous multi-user support must ensure no conflicts arise from shared account use, leveraging session management techniques.
7. Dynamic UI elements must load smoothly and not detract from the overall user experience.
8. The simulator entry page must load quickly, even for users with low-bandwidth connections.
9. Error messages should be designed to be user-friendly, avoiding technical jargon.
10. The map display of forest stands must perform efficiently, allowing smooth interactions even for datasets with many stands.
11. Validation processes must be scalable to handle large datasets without significant performance degradation.
12. Simulation results should be downloadable in a compressed format to accommodate users with limited storage capacity.
13. User account access must include secure authentication mechanisms, such as password hashing and HTTPS encryption.

5.2 Database Design

This section is dedicated to identifying and organising the data that is essential for achieving the project's objectives, while also ensuring its relevance and quality. The data was carefully analysed in order to establish relationships and address potential issues that could impact the simulator's multi-step process.

5.2.1 Initial Data Collection

The initial data from the simulator was established according to the specifications of the stakeholders to ensure alignment with the goals and constraints of forest management. Professors from the Coimbra Agricultural School (ESAC) acted as key stakeholders, offering their forestry science and management expertise to the system's design by identifying critical ecological processes and management variables for the simulator to model.

The simulator assumes the need for a management unit map or landscape map, which represents a forest when together. This serves as a basis for starting the simulation. Each unit of the map can be called a stand and is represented as a polygon with a unique ID, and these boundaries remain consistent throughout the process of generating alternatives. These stands are provided through GIS (Geographic Information System) files.

A management unit or parcel comprises a mosaic of land uses that can fall into the categories listed in Table 5.1.

Table 5.1: Land use types for the simulator

ID	Description
1	Artificialized territories
2	Agriculture
3	Pastures
4	Forests
5	Shrubs
6	Bare or sparsely vegetated areas
7	Wetlands
8	Surface water bodies

Each type of land use may permit specific changes to other types, depending on its characteristics and constraints. These accepted land use transitions are summarised in Table 5.2.

Table 5.2: Accepted land use changes

ID	Current Soil	Land Use Change
1	Artificialized territories	Not allowed
2	Agriculture	Forest
3	Pastures	Forest
4	Forests	Agriculture, Pastures, Shrubs
5	Shrubs	Forest, Agriculture
6	Bare or sparsely vegetated areas	Forest
7	Wetlands	Not allowed
8	Surface water bodies	Not allowed

Furthermore, each polygon representing a forest stand is assigned one of three types of landscape structure, as detailed in Table 5.3. The simulator assumes that polygons classified as matrix will be transformed for forestry use, in line with the validation rules provided above, and will be considered for simulation. Each stand also has a total area and a usable area, which represents the portion of the total area that can be used effectively for simulation.

Table 5.3: Valid landscape types

ID	Landscape Type
1	Resilience Structure
2	Conservation Structure
3	Matrix

Inventory data is available only for polygons classified as forested areas. This data includes key variables such as species composition, occupancy proportion, age class, and quality index. The last three categories will be defined for each species, along with the percentage of occupancy within the polygon.

The type of stand, which significantly influences the simulations, is limited to the categories shown in Table 5.4.

Table 5.4: Type of stands allowed

ID	Stand Type
1	Pure stands
2	Mixed stands

Stand type greatly impacts how simulations are conducted. Pure stands (ID 1) always have a 100% occupancy rate. As a result, simulations for these stands are directly based on per-hectare values obtained from yield tables, ensuring accurate modelling of growth and yield.

For mixed stands (ID 2), the simulation process involves adjusting the yield table values according to the proportion of occupancy of each species. This adjustment is achieved by multiplying the percentage of occupancy by the corresponding values in the tables, allowing an accurate representation of the contribution of each species for the overall value of each stand.

Each polygon/forest stand may have a maximum of three species or groups of species, as listed in Table 5.5.

Table 5.5: Accepted forest species in the simulator

ID	**Designation of Forest Species**
53	*Quercus robur*
54	*Castanea sativa*
59	*Betula alba*
521	*Pinus pinaster*
525	*Pseudotsuga menziesii*

Each species should be associated with a set of yield tables. Yield tables are crucial for forest management, as they provide valuable information on tree growth patterns and potential productivity in different forest stands. Forest managers and practitioners use these models to make decisions about the management of individual forest stands or entire estates, forecast production levels, make commitments to the timber markets, and plan forest operations [91].

Yield classes and volume were estimated using yield tables, which were adjusted to provide volume and thinning estimates at consistent intervals. For each species, three yield classes were considered to provide estimations of the volume for the main crop after thinning and thinning at one given age class: high, medium and low. The yield tables from Diéguez-Aranda *et al.* [92] were used for *Pseudotsuga menziesii*, *Quercus robur*, and *Betula alba*, while the tables from Santos *et al.* [93] and Patrício [94] were applied for *Pinus pinaster* and *Castanea sativa*.

Since yield tables were originally designed for pure stands, their use in mixed stands is limited to mixed stands by patches or groups. Consequently, for each species, it is necessary to define a silvicultural model, which includes specifying the silvicultural system (Table 5.6) as well as the maximum and minimum harvest ages to be considered when generating management alternatives evaluated by the optimisation algorithm. It is acknowledged that both harvest age and silvicultural system can significantly influence the results.

Table 5.6: Silvicultural system for the simulator

ID	**System**
1	Clearcut
2	Shelterwood

The proportion of occupancy (PO) is expressed as a percentage of the area occupied by each species or group of species in the stand, as indicated in the inventory data file.

To ensure accuracy, several validations are applied to the species occupancy data. First, the total proportion of occupancy, which is the sum of the percentages for up to three species, must be equal to 100%. Secondly, no individual species can have an occupancy percentage below 25%. These rules guarantee the consistency and validity of the data used for the simulations.

Each polygon and each species must also be assigned an age class. Additionally, a quality index is defined for each species, calculated in one of two ways: (1) from inventory data or (2) through a qualitative estimate of the quality class. Details of these calculations will be provided later.

5.2.2 Database Structure

The database that supports the simulator was designed with the objective of efficiently storing and organising the necessary data. The database incorporates tables for the storage of user-uploaded files, in addition to the essential data concerning the simulator logic. Table 5.7 presents a summary of the database tables and Figure 5.1 represents the database schema.

Table 5.7: Database for the simulator main structure

Table Name	**Description**
`StandData`	Stores polygons representing forest stands
`InventoryData`	Contains inventory data for forested areas
`YieldSpecies`	Yield tables for each species
`ForestSpecies`	Contains forest species identification
`StandComposition`	Stores compositions for forest stands
`LandscapeTypes`	Stores landscape types
`LandUse`	Specifies land use types and allowed land-use transitions
`SilviculturalSystem`	Stores the silvicultural system, defining the type of harvest

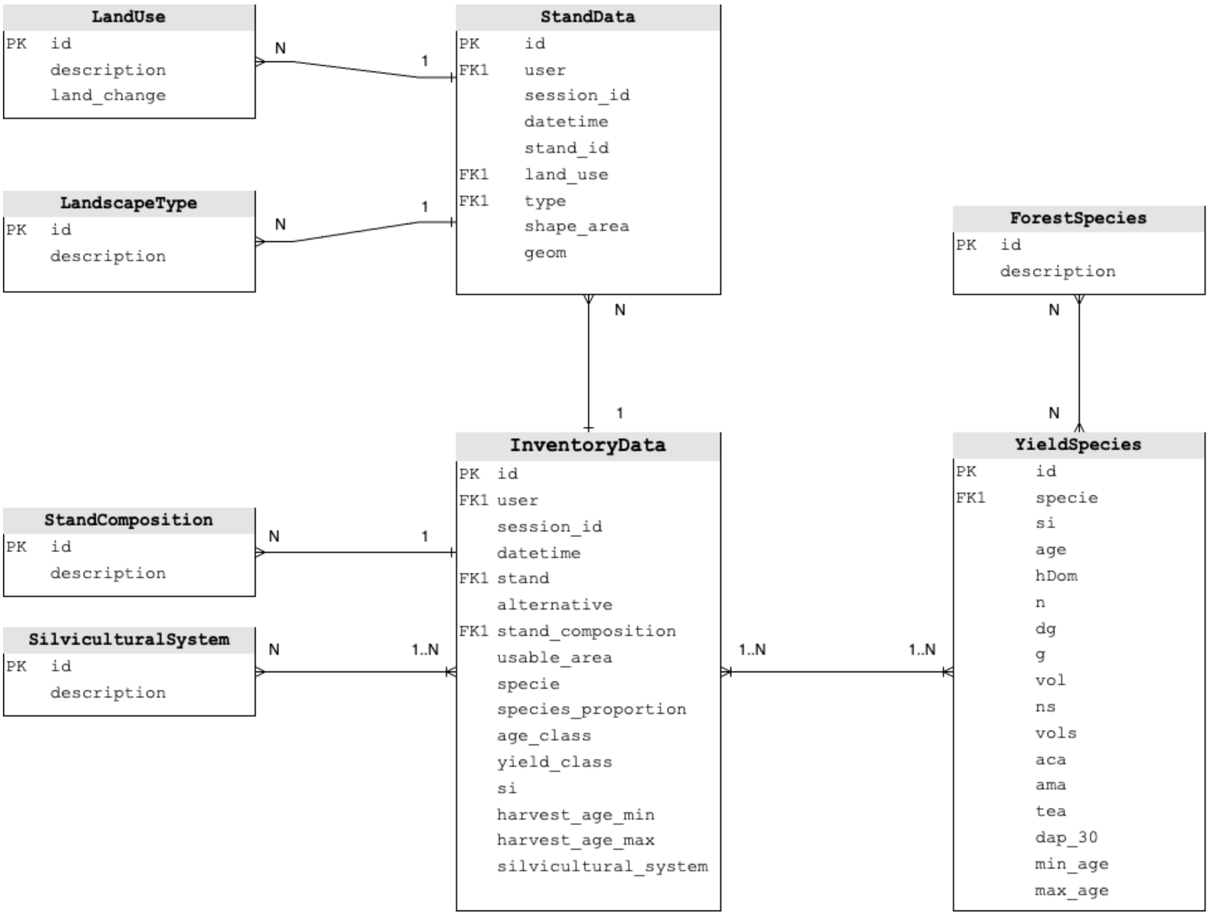


Figure 5.1: Database schema for the forest management simulator

The database schema illustrates the relationships established between the database tables to support the entire simulation process.

The `LandUse` table is used to categorise land use in a given area and to track allowed changes in land use over time, according to Table 5.1. Each record in `LandUse` can be associated with multiple forest stands recorded in the `StandData` table (1:N relationship). Its attributes are:

- `id`: The primary key, uniquely identifying each land use record.
- `description`: A textual description of the type of land use, such as "agriculture," "forest reserve", or "pastures".
- `land_change`: A field to track whether or not it allows changes in land use.

The `LandscapeType` table categorises the physical or ecological component of the landscape as described in Table 5.3. This table supports a one-to-many relationship with the `StandData` table, as a single landscape type can be applied to multiple forest stands. Its attributes include the following.

- `id`: The primary key, uniquely identifying each type of landscape.
- `description`: A textual description of the type of landscape.

The `StandData` table acts as the main source of information for distinct forest stands, which are forest sections typically managed as unified units. This table is central to the schema, linking data on land use (`LandUse`), landscape types (`LandscapeType`), and spatial characteristics.

Each forest stand recorded in `StandData` can have multiple associated entries in the `InventoryData` table (1:N relationship), according to the option of adding more combinations of species of stands. The attributes are as follows:

- `id`: The primary key, uniquely identifying each stand.
- `user`: The identifier for the user who recorded the data.
- `session_id`: A session identifier to track the context of data collection.
- `datetime`: A timestamp indicating when the data was recorded.
- `stand_id`: A unique identifier for the forest stand within the system.
- `land_use`: A foreign key referencing the `LandUse` table, linking the stand to its designated land use type.
- `type`: A foreign key referencing the `LandscapeType` table, specifying the landscape type for the stand.
- `shape_area`: A numeric field representing the total area of the stand.
- `geom`: Spatial geometry data, such as georeferenced polygons used to represent standsin a map.

The `InventoryData` table stores detailed records of the inventory of forest stands, capturing essential information for defining harvest ages and predictions of forest yields. Each inventory entry links to a specific stand recorded in the `StandData` table (N:1 relationship). Furthermore, each inventory entry can reference multiple entries in the `YieldSpecies` table (1:N relationship), which provides detailed growth and yield data for individual species. The `InventoryData` table also connects to the `StandComposition` and `SilviculturalSystem` tables, allowing classification of the stand's composition and management practices. Its attributes include:

- `id`: The primary key, uniquely identifying each inventory record.
- `user`: The user responsible for entering the data.
- `session_id`: The session identifier for tracking the data collection context.
- `datetime`: A timestamp for when the inventory data was recorded.
- `stand`: A foreign key linking to the `StandData` table, associating the inventory with a specific stand.
- `alternative`: An optional field for counting the number of alternative management scenarios.
- `stand_composition`: A foreign key linking to the `StandComposition` table, specifying the composition of the stand.
- `usable_area`: The percentage of usable area of the stand.
- `specie`: A foreign key linking to the `ForestSpecies` table, identifying the species present in the stand.
- `species_proportion`: The proportion of the area of the stand occupied by each species.
- `age_class`: The age classification for each species.
- `yield_class`: A volume productivity classification for the stand or species.
- `si`: Site index, a measure of forest productivity specific for each species and each site conditions.
- `harvest_age_min`: The minimum age at which the stand or species can be harvested.
- `harvest_age_max`: The maximum age at which the stand or species can be harvest
- `silvicultural_system`: A foreign key referencing the `SilviculturalSystem` table, defining the forestry system applied to the stand that determines the type of harvest applied

The `StandComposition` table describes the overall composition of forest stands, Table 5.4.

Each composition can be linked to multiple records in the `InventoryData` table (1: N relationship), allowing the classification of stands based on their ecological or species composition. It has the following attributes:

- `id`: The primary key, uniquely identifying each type of stand composition.
- `description`: A textual description of the composition, such as "mixed coniferous and deciduous" or "monoculture of pine."

The `SilviculturalSystem` datatable is a representation of Table 5.6. Each silvicultural system can be associated with multiple records in the `InventoryData` table (1:N relationship). Its attributes contain:

- `id`: The primary key, uniquely identifying each silvicultural system.
- `description`: A textual description of the system.

The `ForestSpecies` table contains basic information about tree species found in forest stands (Table 5.5). Each species can be referenced by multiple records in the `YieldSpecies` table (1:N relationship), where growth and yield data are recorded. Its attributes are:

- `id`: The primary key, uniquely identifying each species.
- `description`: A textual description of the species, often including its scientific or common name.

The `YieldSpecies` table captures detailed data on the growth and yield characteristics of tree species. This table supports advanced analysis of species productivity and links to `ForestSpecies` (N:1 relationship) and `InventoryData` (1:N relationship). Its attributes include the following.

- `id`: The primary key, uniquely identifying each yield record.
- `specie`: A foreign key linking to the `ForestSpecies` table, specifying the species.
- `si`: Site Index.
- `age`: Age class of the species.
- `hDom`: Dominant height of the species at a given age class.
- `N`: Number of trees per hectare.
- `dg`: Diameter (cm) at breast height (DBH) of the species.
- g: Basal area (m2)
- `vol`: Volume of the main crop after thinning (m3).
- `ns`: Number of trees from thinning
- `vols`: Volume from thinning (m3)
- `aca`: Annual volume increment
- `ama`: Mean volume increment, calculated dividing volume by the age of the stand.

- `tea`: Age class in which the aca equals ama.
- `dap_30`: Age classe at which the trees have a diameter at breast height (DBH) greater than 30 cm.
- `min_age`: Minimum age for harvest simulation.
- `max_age`: Maximum age for harvest simulation.

5.3 Workflow and User Interaction

This section is dedicated to illustrating the interactions between the front-end and user actions with the back-end components of the simulator through a cross-functional flowchart. The diagram 5.2 presents a visual representation of the sequence of actions and processes that occur within each phase, offering a complete understanding of the manner in which the back-end processes user input.

5.3.1 Step 1: Import Forest Stands

Users begin by accessing the dedicated page for all simulators and selecting the forest management simulator. If the user is logged in, they are redirected to the simulator's welcome page; otherwise, they are prompted to log in or register. The welcome page provides an overview of the usage rules, access to sample files, and an option to start the simulation. Upon starting the simulation, the user is taken to the first step, where the simulator begins to collect the required input data.

In the first step, users are required to upload a GIS file containing information about forest stands. If the file is successfully imported and validated, a success message is displayed, allowing the user to proceed to the next step. If errors are detected, the user is prompted to re-upload the file to ensure compliance with the system requirements.

While users interact with the front-end, the back-end performs several critical operations to process the uploaded file. First, the system extracts and separates the relevant data from the GIS file, temporarily saving it in designated folders. Once the data are extracted, the back-end reads the files and validates the content. This includes verifying the number of forest stands, ensuring that the file was imported correctly, and checking for compliance with the required formats and constraints. If the data passes validation, it is imported into a database table called `StandData`. After successfully populating the database, the temporary files are deleted to optimise the usage of the system resources.

The back-end then generates a status message, either confirming successful import or indicating specific errors encountered during the process. This message is sent to the front end and displayed to the user. If the process is successful, the front-end prompts the user to decide whether to proceed to the next step. Upon confirmation, the back-end manages the transition, redirecting the user to the second step of the simulation. This coordinated interaction between the front-end and back-end ensures that data integrity is preserved while providing users with clear and actionable feedback throughout the process.

Figure 5.3 demonstrates the simulator's operation during this phase, while Figure 5.4 displays the status messages that the simulator presents to the user following each step.

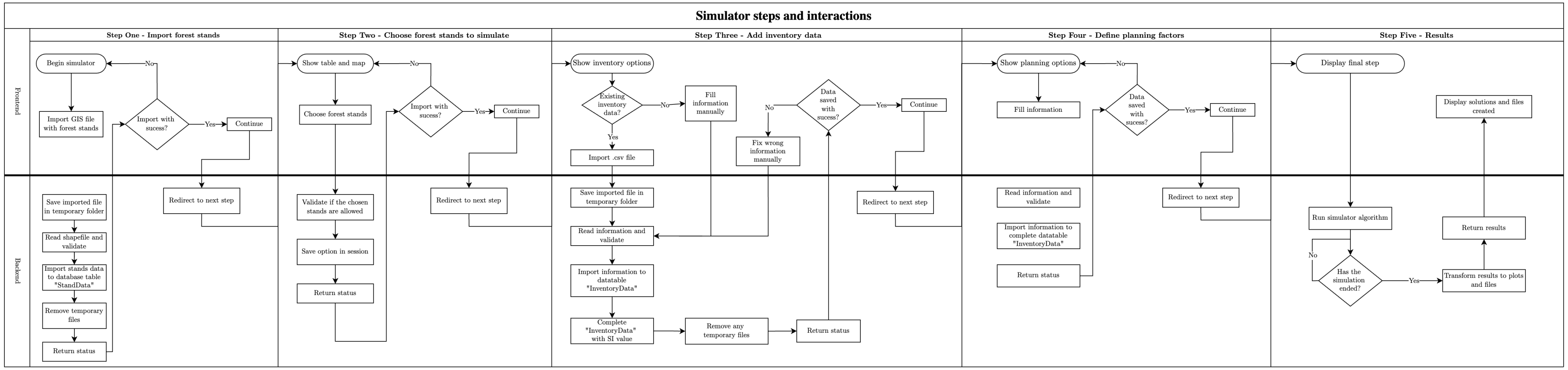


Figure 5.2: Simulator steps and interactions with the Front-End and Back-End of the platform

Importar dados

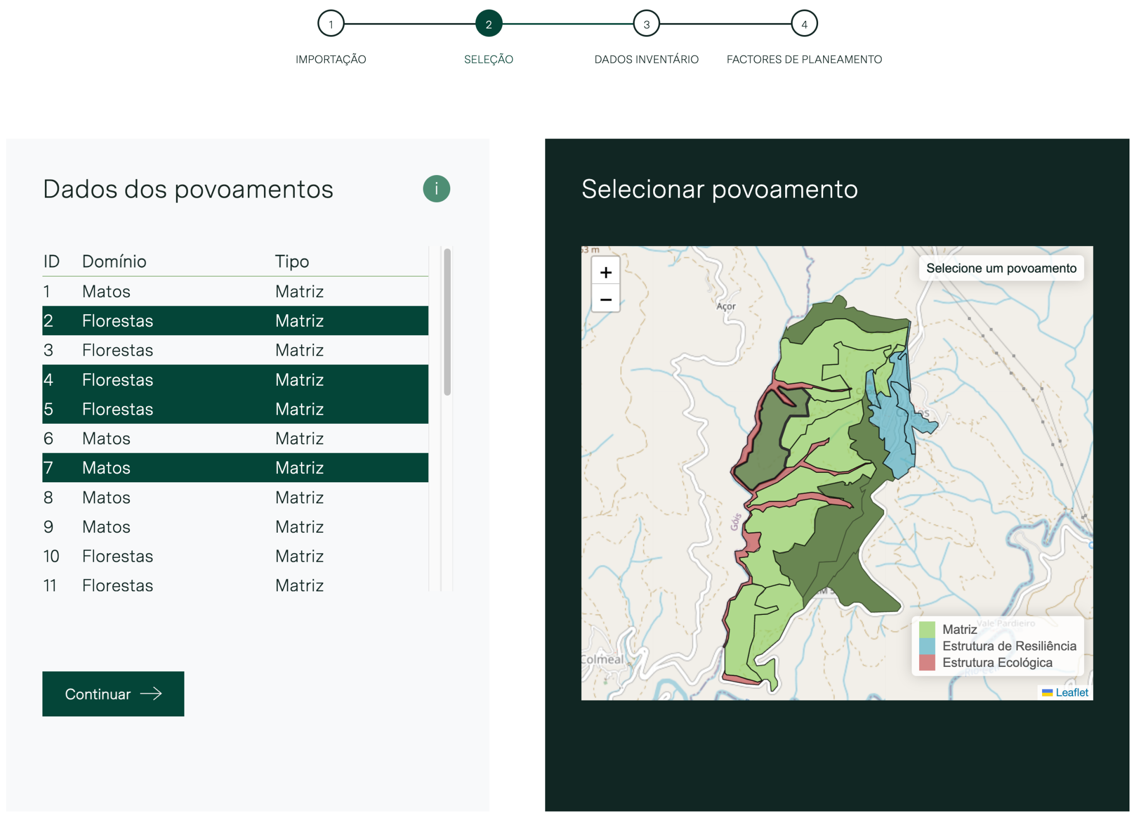
Ficheiro no formato .zip

povoamentos.zip

Figure 5.3: Webpage example for the first step of the simulator

Ficheiro importado com sucesso!

Figure 5.4: Status message page for the simulator



Dados dos povoamentos

ID	Domínio	Tipo
1	Matos	Matriz
2	Florestas	Matriz
3	Florestas	Matriz
4	Florestas	Matriz
5	Florestas	Matriz
6	Matos	Matriz
7	Matos	Matriz
8	Matos	Matriz
9	Matos	Matriz
10	Florestas	Matriz
11	Florestas	Matriz

Selecionar povoamento

Selecione um povoamento

🟩 Matriz
🟦 Estrutura de Resiliência
🟥 Estrutura Ecológica

Figure 5.5: Form for the second step of the simulator

5.3.2 Step 2: Selection of Forest Stands

In step two, users are presented with the uploaded forest stands, displayed both as a map and a table, providing a clear and structured view of the data. Users are required to select the stands they wish to include in the simulation. Once the selection is submitted, the back-end validates the chosen stands against the simulator's logic and requirements to ensure compliance with the expected parameters.

Following this validation, the selected stands are saved within the user's session. This approach ensures that the initial forest stand selections remain intact, even if the user decides to revisit step three or step four at a later stage. Upon successful validation, the back-end returns a status message indicating the outcome. If the stands are validated and saved correctly, the user is permitted to proceed to the next step. In the event of an error, the system prompts the user to restart step two, ensuring that the selection aligns with the required criteria.

Once the validation process is completed and the stands are correctly selected, the user is prompted to proceed to step three, which involves adding inventory data.

Figure 5.5 illustrates the process of selecting hypothetical forest stands for use in the simulator.

5.3.3 Step 3: Inclusion of Inventory Data

In step three, users are instructed to provide inventory data. This can be done by importing a data file or manually entering the information at the time. This step ensures that essential data about each selected forest stand are correctly integrated into the system.

If users opt to import a `.csv` file, the back-end saves the file in a temporary folder for processing. In contrast, if users choose to input the data manually, a form is displayed, allowing them to provide the necessary details. Once the form is completed, the back-end reads the submitted data and verifies that all required information is included. After successful validation, the inventory data are associated with the corresponding stands and stored in the database table `InventoryData`.

During this step, the back-end also refines the inventory data by calculating and assigning additional attributes. Specifically, it determines the Site Index (SI) for each forest species within a stand by referencing the yield tables. This calculation is based on the species and their respective yield class, ensuring that all relevant metrics are incorporated into the simulation.

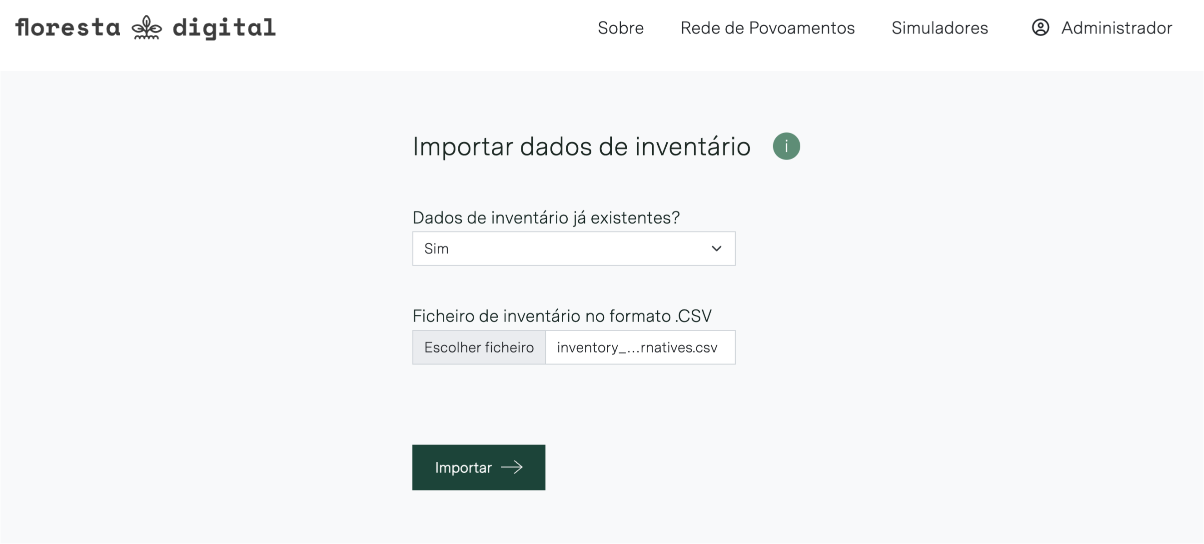
After completing data validation and augmentation, the system checks for any remaining temporary files and removes them to optimise resource management. Then a status message is returned, indicating whether the step was completed successfully or if further action is required.

If errors are detected during the validation process, the user is prompted to correct them through a form. This form highlights the fields containing errors and provides clear guidance on how they should be resolved. Once all issues are resolved and the data is saved correctly, the user is allowed to proceed to Step 4. This structured process ensures the accuracy and completeness of inventory data while maintaining a seamless workflow for the user.

Figure 5.6 illustrates the interface used for importing the inventory data file. While Figure 5.7 represents the option to manually add inventory data.

5.3.4 Step 4: Defining Planning Factors

In this step, the users define the planning factors necessary for the simulation. Initially, they are asked to specify the overall planning time frame and the planning period, which can be either 5 or 10 years. Following this, users must define the minimum and maximum harvest ages for each species within each selected stand. Additionally, users can specify the type of silvicultural system applicable, such as clearcut or shelterwood, for greater flexibility in the simulation setup.

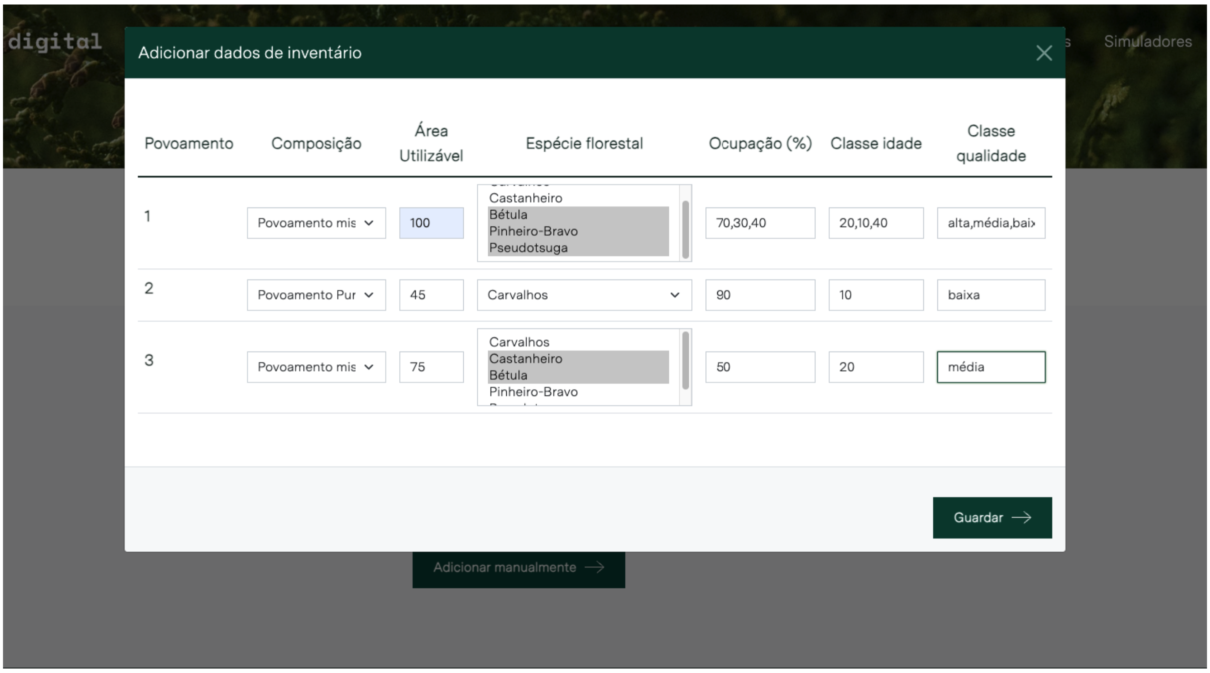


Importar dados de inventário

Dados de inventário já existentes?
Sim

Ficheiro de inventário no formato .CSV
inventory_...rnatives.csv

Figure 5.6: Webpage example for importing inventory data for the simulator



Adicionar dados de inventário

Povoamento	Composição	Área Utilizável	Espécie florestal	Ocupação (%)	Classe idade	Classe qualidade
1	Povoamento mis	100	Castanheiro Bétula Pinheiro-Bravo Pseudotsuga	70,30,40	20,10,40	alta,média,bai
2	Povoamento Pur	45	Carvalhos	90	10	baixa
3	Povoamento mis	75	Carvalhos Castanheiro Bétula Pinheiro-Bravo	50	20	média

Figure 5.7: Webpage example for adding inventory data manually

For species with an age of zero, the system interprets this as an indication that the forest stand can accommodate alternative compositions made up of species combinations. In such cases, users have the option to add new species combinations by accessing a form. This form allows users to define the new composition of the stand by selecting the species, specifying the occupancy rate and yield class of the stand, while modifying the overall composition for the specific stand.

Once all fields in the form have been completed, the back-end validates the data to ensure that it adheres to the simulation logic and requirements. Upon successful validation, the system updates the relevant attributes in the database table `InventoryData`. Then a status message is returned, indicating whether the operation was successful. If successful, the user is allowed to proceed to the final step of the simulation.

During the transition to the next step, a loading page is displayed while the system processes the data and prepares to show the simulation results. Figure 5.8 illustrates the simulator in this phase, depicting a completed setup with a forest scenario sample.

5.3.5 Step 5: Results

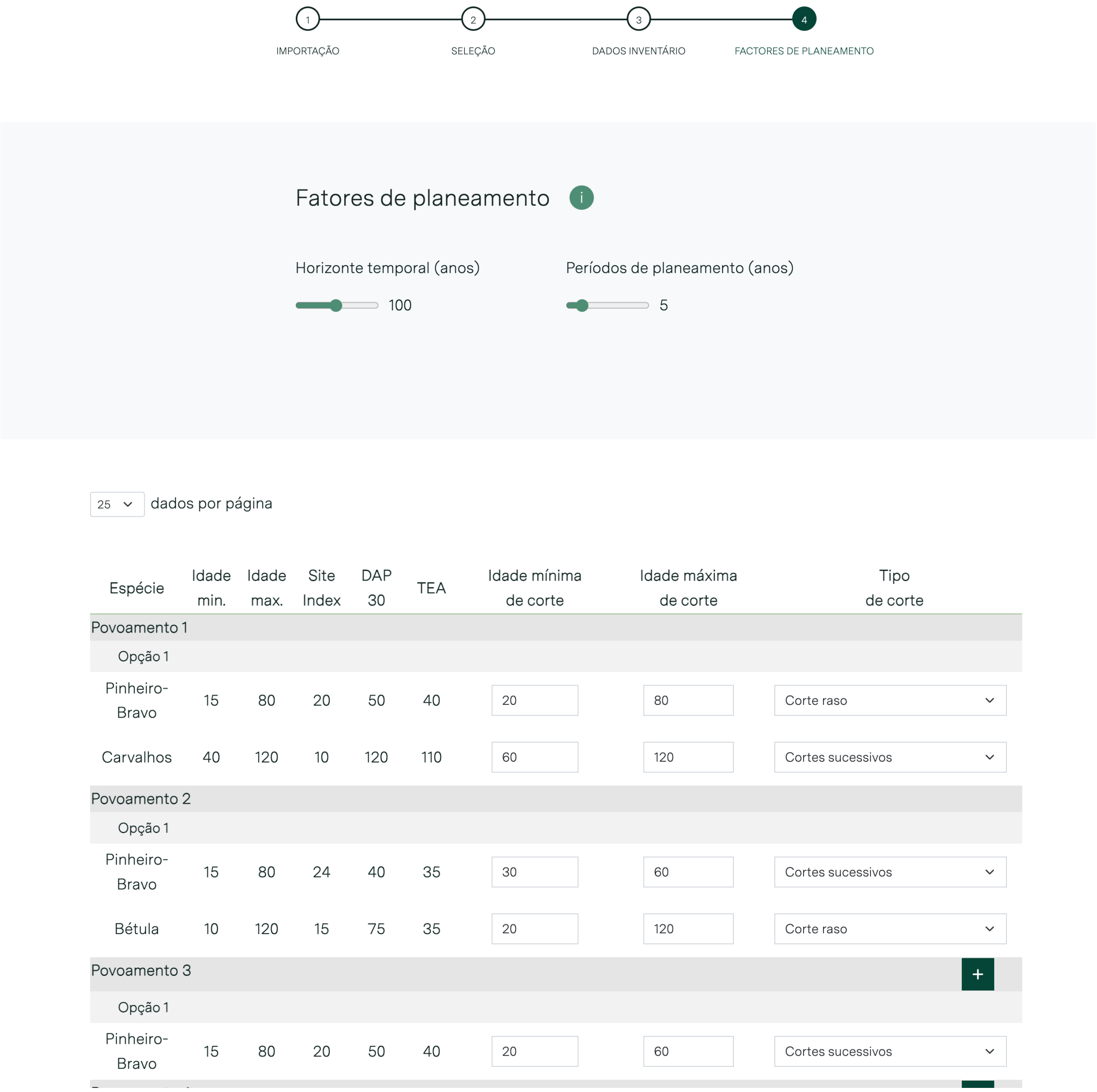
In the final phase, the simulator executes a specialized algorithm designed to evaluate all possible combinations of forest species and their respective harvest ages. The objective of the algorithm is to identify the Pareto optimal solutions that balance the competing goals defined by the simulation. This optimisation ensures that the generated solutions are efficient and relevant to the forest management objectives.

Once the algorithm is successfully completed, the results are processed and transformed into visual outputs, including graphs, and are formatted for download in Excel files. These outputs are then displayed in the front-end interface, providing users with clear and actionable information.

On the results page, users can choose to analyse the outcomes directly through interactive solution plots, which allow detailed exploration of the simulation's findings, Figures 5.9 – 5.10. Alternatively, users may export the results into an Excel file containing the solutions for further analysis.

In addition to the algorithm's core results, the exported files include supplementary data detailing the forest species volumes for each planning period. This additional information, derived from intermediate calculations used during the simulation, is a valuable resource for forest managers, as it aligns with metrics utilised in previous studies.

Figures 5.9 and 5.10 present the results of the simulator based on a hypothetical forest scenario.



Fatores de planeamento

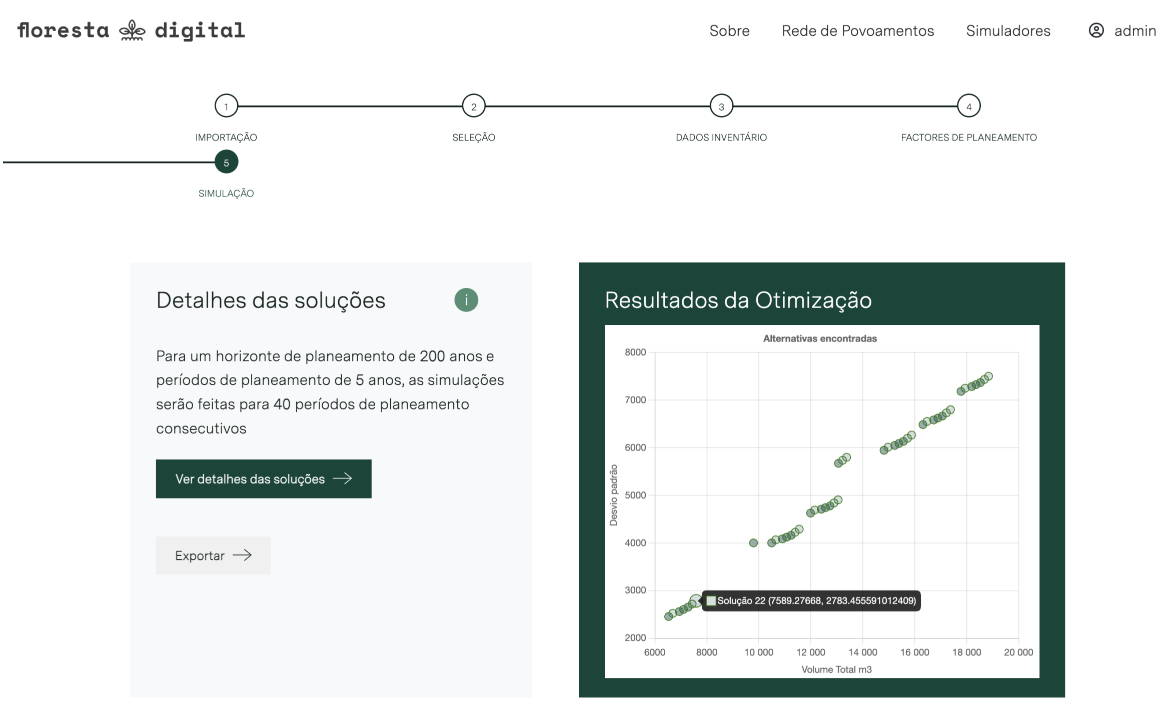
Horizonte temporal (anos): 100

Períodos de planeamento (anos): 5

25 dados por página

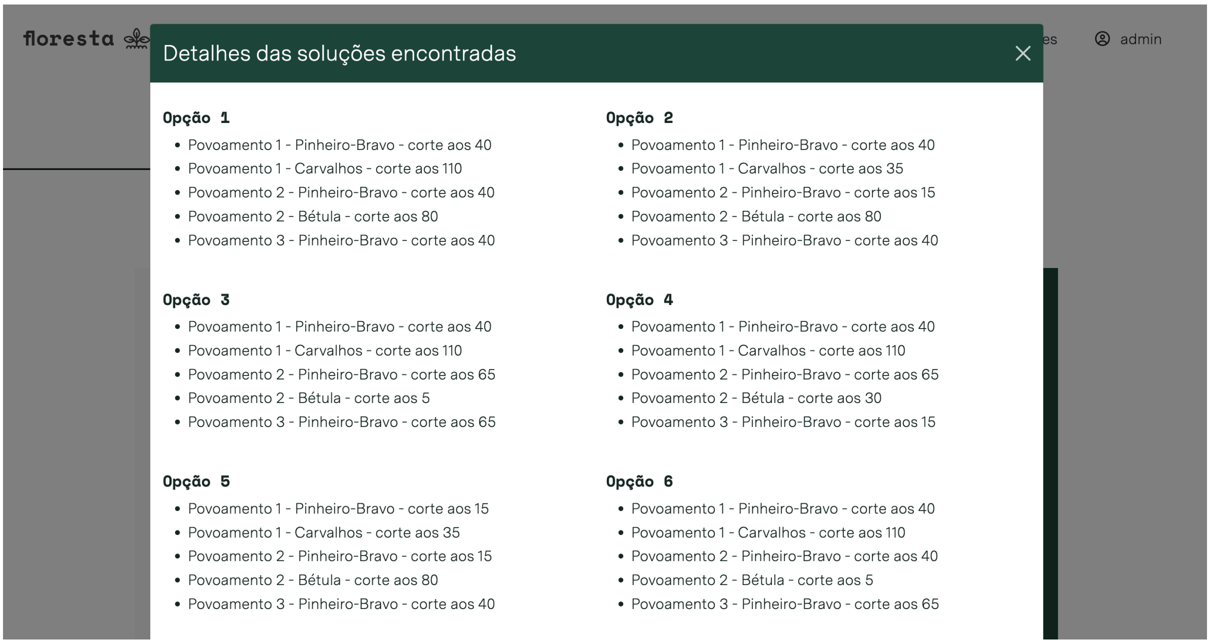
Espécie	Idade min.	Idade max.	Site Index	DAP 30	TEA	Idade mínima de corte	Idade máxima de corte	Tipo de corte
Povoamento 1								
Opção 1								
Pinheiro-Bravo	15	80	20	50	40	20	80	Corte raso
Carvalhos	40	120	10	120	110	60	120	Cortes sucessivos
Povoamento 2								
Opção 1								
Pinheiro-Bravo	15	80	24	40	35	30	60	Cortes sucessivos
Bétula	10	120	15	75	35	20	120	Corte raso
Povoamento 3								
Opção 1								
Pinheiro-Bravo	15	80	20	50	40	20	60	Cortes sucessivos

Figure 5.8: Webpage example for step four of the simulator



Para um horizonte de planeamento de 200 anos e períodos de planeamento de 5 anos, as simulações serão feitas para 40 períodos de planeamento consecutivos

Figure 5.9: Simulator results for forest management alternatives



Opção 1
- Povoamento 1 - Pinheiro-Bravo - corte aos 40
- Povoamento 1 - Carvalhos - corte aos 110
- Povoamento 2 - Pinheiro-Bravo - corte aos 40
- Povoamento 2 - Bétula - corte aos 80
- Povoamento 3 - Pinheiro-Bravo - corte aos 40

Opção 2
- Povoamento 1 - Pinheiro-Bravo - corte aos 40
- Povoamento 1 - Carvalhos - corte aos 35
- Povoamento 2 - Pinheiro-Bravo - corte aos 15
- Povoamento 2 - Bétula - corte aos 80
- Povoamento 3 - Pinheiro-Bravo - corte aos 40

Opção 3
- Povoamento 1 - Pinheiro-Bravo - corte aos 40
- Povoamento 1 - Carvalhos - corte aos 110
- Povoamento 2 - Pinheiro-Bravo - corte aos 65
- Povoamento 2 - Bétula - corte aos 5
- Povoamento 3 - Pinheiro-Bravo - corte aos 65

Opção 4
- Povoamento 1 - Pinheiro-Bravo - corte aos 40
- Povoamento 1 - Carvalhos - corte aos 110
- Povoamento 2 - Pinheiro-Bravo - corte aos 65
- Povoamento 2 - Bétula - corte aos 30
- Povoamento 3 - Pinheiro-Bravo - corte aos 15

Opção 5
- Povoamento 1 - Pinheiro-Bravo - corte aos 15
- Povoamento 1 - Carvalhos - corte aos 35
- Povoamento 2 - Pinheiro-Bravo - corte aos 15
- Povoamento 2 - Bétula - corte aos 80
- Povoamento 3 - Pinheiro-Bravo - corte aos 40

Opção 6
- Povoamento 1 - Pinheiro-Bravo - corte aos 40
- Povoamento 1 - Carvalhos - corte aos 110
- Povoamento 2 - Pinheiro-Bravo - corte aos 40
- Povoamento 2 - Bétula - corte aos 5
- Povoamento 3 - Pinheiro-Bravo - corte aos 65

Figure 5.10: Detailed results for forest management alternatives

5.4 Summary

In conclusion, this chapter has provided an overview of the work completed for the initial module of the simulator, namely, the interface design and data preparation. With these essential steps completed, the subsequent sections will address the second component of the project, specifically the implementation of a multi-objective optimisation algorithm.

6 INTEGRATION OF AN OPTIMISATION ALGORITHM

This chapter is dedicated to the integration of a multi-objective optimisation algorithm into the forest management simulator, which forms the second module of the simulator. The inclusion of this algorithm enables the simulator to effectively generate a Pareto-optimal extensive front, providing several solution sets for the optimisation problem.

The chapter is divided into the following sections. Section 6.1 presents the mathematical formulation of the optimisation problem. Section 6.2 outlines the multi-objective algorithm strategy. Section 6.3 provides a sample scenario to demonstrate the algorithm's evaluation and Section 6.4 examines the algorithm's performance.

The following sections are based on the findings of a published paper that resulted from this project work [25]. The paper demonstrates the effective application of the NSGA-II algorithm to a forest management case study, demonstrating its capacity to tackle multi-objective optimisation problems and making significant contributions to sustainable forest management.

6.1 Formulation of the Optimization Problem

This project addresses an issue concerning a forest with various and distinct stands, each hosting up to three species of differing ages, both current and at harvest. Each stand can accommodate various compositions, enabling forest planners to add new species if needed.

The combinatorial problem analysed here aims to find the optimal combinations of stands and harvest ages to achieve two objectives simultaneously: maximising the total timber harvest and minimising the standard deviation of the total timber harvest for each period.

Optimisation goals focus on treating each stand with consideration for the collective objectives of the entire forest. The mathematical formulation of the problem may be described according to Equations 6.1 – 6.2.

$$\begin{aligned} &\text{Maximize} \quad V_{\text{total}} = \sum_{n=1}^{N} \sum_{p=1}^{P} \sum_{t=1}^{T} V_{npt} \\ &\text{Subject to} \quad V_{npt} \geq 0, \\ &\qquad\qquad\quad \forall\, n = 1, \ldots, N,\ p = 1, \ldots, P,\ t = 1, \ldots, T \end{aligned} \tag{6.1}$$

$$\begin{aligned} &\text{Minimize} \quad S_{\text{total}} = \sqrt{\frac{\sum_{t=1}^{T} (X - \bar{x})^2}{T - 1}} \\ &\text{Subject to} \quad \forall\, t = 1, \ldots, T \end{aligned} \tag{6.2}$$

where
V_{total} = volume harvested during the planning horizon for all stands *N*;
N = number of stands;
P = number of species;
T = number of harvest periods;
V_{xtc} = timber volume harvested during period *t* for species *x* of stand *n*;
S_{total} = standard deviation value during the planning horizon for all stands *N*;
X = stand's timber volume for period *t*;
$\bar{x}$ = mean of the sample for period *t*.

The timber volumes for each stand *n* can be obtained using Equation 6.3.

$$V_{npt} = S_{area} \times P_{proportion} \times (v_{stand} + \sum_{i=1}^{n} v_{thinnings}) \tag{6.3}$$

where
S_{area} = proportion of usable area in stand *n*;
$P_{proportion}$ = ratio of species *p* in stand *n*;
v_{stand} = volume of the stand for period *t*;
$v_{thinnings}$ = volume obtained from thinning.

This bi-objective optimisation problem is complemented by a constraint where the total volume cannot be zero, as this represents an invalid solution: $V_{total} > 0$. Moreover, in the course of validating the solutions generated by the algorithm, any solution that exhibited harvest age values not within the specified ranges was considered to violate the constraints, according to the Equation 6.1.

6.2 Multi-Objective Algorithm Strategy

Preliminary experiments showed enough variable results to allow ongoing improvements to the second simulator module. At an early stage, modifications were made to enhance the primary structure of the problem, and subsequently adjustments were introduced to optimise the genetic operators. These refinements improved the performance of the algorithm and reduced its computational demands. Reeves *et al.* [44] highlighted that Genetic Algorithms are naturally adaptable, allowing easy modifications for changes in the original problem.

6.2.1 Chromosome Representation

The Non-Dominated Sorting Genetic Algorithm (NSGA-II), implemented through the PyMoo package in Python [85], was used to find Pareto-optimal solutions, where improving one objective cannot occur without worsening another.

NSGA-II is a state-of-the-art Multi-Objective Genetic Algorithm (MOGA) designed to overcome the limitations of classical optimisation methods, such as high computational complexity, non-elitism, and the need to specify a sharing parameter [24]. By integrating elitism, NSGA-II preserves the best solutions from the previous generation while employing genetic operators, mutation, and crossover to generate a new population. This approach not only accelerates convergence towards the optimal solution, but also enhances the overall efficiency of the search process.

This algorithm effectively manages optimisation constraints and maintains diversity in the population by using a crowded comparison operator. The solutions are initially ranked according to dominance, and then sorted according to crowding distance, which contributes to an efficient ranking system that reduces computational complexity [22, 95].

The approach carried out here highlights the efficient use of multi-objective optimisation algorithms to meet specific requirements and goals. Even in the face of strict limits, NSGA-II is able to generate various solutions for a forest management problem.

For the application of NSGA-II, the problem was encoded as follows. The Genetic Algorithm uses a chromosome of length L, where L is determined by the number of compositions, the number of species in each composition, and the defined harvest age periods for each species.

Specifically, the length of the chromosome L is given by Equation 6.4.

$$L = C + \left(\sum_{j=1}^{C} I \right) \times T \tag{6.4}$$

where
C =number of compositions;
I = number of species in each composition;
j = number of harvest periods.

For example, consider the following values:

- Number of compositions: $C = 3$;
- Species per composition:
  - Composition C_1 has $I = 2$;
  - Composition C_2 has $I = 3$;
  - Composition C_3 has $I = 1$.
- Number of harvest periods: $T = 10$.

Using the formula designated as Equation (6.4), the following calculation is obtained:

Sum of species across all compositions:

$$\sum_{j=1}^{C} I = C_1 + C_2 + C_3 = 2 + 3 + 1 = 6 \text{ species}$$

Chromosome length calculation:

$$L = C + 6 \times T = 3 + (6 \times 10) = 3 + 60 = 63 \text{ genes}$$

As a result, for this test configuration, the length of chromosome L is 63.

Each cell in this chromosome represents a gene, allowing the algorithm to evaluate and optimise based on the encoded information. Although problems typically solved with the PyMoo package are continuous in nature, it is also possible to use other types of variables [85]. In this case, a binary gene type is used. The first set of cells on the chromosome, which represent the stand's alternatives, are subjected to a constraint that allows only one composition per stand to be selected. This constraint is applied across all stands.

Figure 6.1 illustrates the arrangement of genes within the problem chromosome, as well as the application of genetic operators such as crossover and mutation.

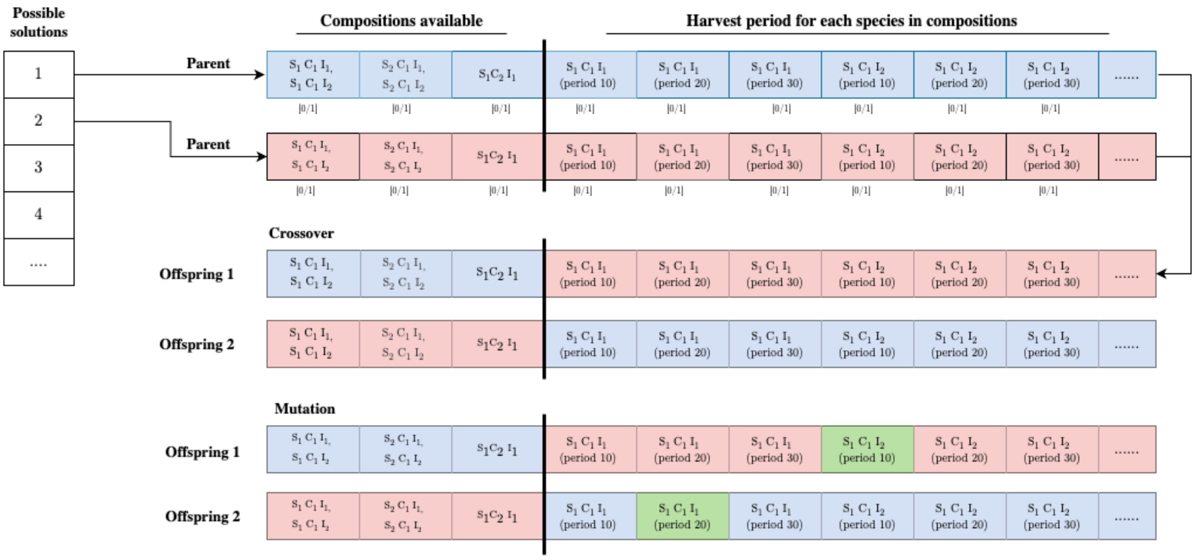


Figure 6.1: Data encoding for the NSGA-II algorithm.

Each gene in Figure 6.1 is encoded by the combination $S_iC_jI_k$:

- S_i represents stand i, where $1 \leq i \leq m$, with m being the total number of stands
- C_j represents stand composition j, where $1 \leq j \leq n$, with n being the maximum number of stand compositions
- I_k represents species k, where $1 \leq k \leq r$, with r being the maximum number of species

In this context

- i, j, and k are index variables used to identify the stands, compositions, and species, respectively.
- m, n, and r are constants denoting the maximum number of stands, compositions, and species in the system.

Subsequently, for each species present in the compositions, the corresponding gene is repeated t times, where T represents the number of periods.

For each group of periods, only one cell can represent the harvest age, ensuring that only one period is designated as the harvest age for that species according to the age range previously defined.

In the course of developing and implementing the NSGA-II algorithm to address this particular problem, it became evident that the scale of the problem posed significant challenges. The complexity of the problem was evidenced by the fact that each chromosome, for a simple forest scenario, consisted of more than 100 genes. This complexity proved detrimental to the algorithm, which was unable to achieve an acceptable ratio of valid to invalid solutions.

This issue was particularly notable during the initial stages of the algorithm's execution, with the need for more than 1000 iterations before the algorithm reached an optimal solution.

Therefore, it was established that the algorithm required modification in order to enhance the generation of valid solutions.

This custom NSGA-II algorithm differs from those commonly used in forest management problems, as it accommodates the presence of composition options for each stand [20], harvest-age limits, and a custom initial population and mutation operator. Furthermore, the algorithm is integrated into a unique web-based platform, facilitating broad accessibility for any user.

The following sections provide a comprehensive analysis of the alterations made to the original NSGA-II algorithm, as outlined by the pseudocode in Algorithm 1. Additionally, the code listing 1 provides the primary setup for this custom algorithm within the Pymoo framework.

Algorithm 1 Custom NSGA-II Procedure

```
Require: $N'$, $g$, $f_k(X)$        ▷ $N'$ members evolved over $g$ generations to solve $f_k(X)$
Initialize Population $P_0$
Generate custom population of $N'$ based on problem constraints
Calculate Objectives $V_{total}$ and $S_{total}$
Evaluate Constraint ($V_{total} > 0$)
Assign Rank based on Pareto dominance
Generate Child Population:
    Binary Tournament Selection
    Crossover and custom Mutation
for $i = 1$ to $g$ do
    for each Parent and Child in Population do
        Assign Rank based on Pareto dominance
        Generate sets of non-dominated solutions
        Determine Crowding distance
        Add solutions to next generation starting from the first front until $N'$ individuals
    end for
    Select points on the lower front with high crowding distance
    Create next generation
        Binary Tournament Selection
        Crossover and custom Mutation
end for
Output Results
    Extract and plot best solutions
    Plot convergence and Pareto front
```

Listing 1 Custom NSGA-II algorithm configuration with Pymoo

```
algorithm = NSGA2(
    pop_size=config['pop_size'], # Population size from configuration file
    sampling=CustomSampling(
        all_alternatives, # List of possible alternative stands combinations
        n_var_genes, # Chromosome size
        n_stands_alternatives, # Total number of possible stands combinations
        n_periods_species, # Total of planning periods per species
        n_periods, # Number of planning periods
        cut_periods # Harvest periods interval
    ),
    selection=TournamentSelection(func_comp=binary_tournament),
    crossover=SBX(
        prob=config['crossover_prob'], # Crossover probability from
        ↪  configuration file
        vtype=float,
        repair=RoundingRepair()
    ),
    mutation=CustomMutation(
        all_alternatives,
        n_var_genes,
        n_stands_alternatives,
        n_periods_species,
        n_periods,
        prob=config['mutation_prob']  # Mutation probability from configuration
        ↪  file
    ),
    eliminate_duplicates=True # Option to remove duplicated solutions
)
```

6.2.2 Creation of an Initial Population

A reduction in population size can result in insufficient coverage of the solution space, which can lead to inadequate exploration and an increased risk of premature convergence. On the other hand, an excessively large population can result in a significant increase in computational costs without a corresponding improvement in solution quality [44].

Preliminary tests showed that populations with fewer than 50 individuals exhibited a lack of diversity and frequently converged prematurely, resulting in an insufficient variety of solutions. In contrast, populations of over 500 chromosomes exhibited a considerable increase in computational costs without a corresponding improvement in diversity or the number of solutions, compared to a population size of approximately 200.

To further enhance the performance of the original NSGA-II, a customised operator was developed to generate the initial population (lines 1–2 of the Algorithm 1), replacing the *float random sampling* provided by the PyMoo framework for NSGA-II [96], as shown in lines 3–10 of the code example 1. This approach assigns random binary values to each gene to form a chromosome, guaranteeing validity by ensuring that only one harvest age is selected for each species, rather than multiple. This effectively addresses a key issue in the original NSGA-II approach, where a higher proportion of invalid solutions were generated than valid ones.

6.2.3 Fitness Evaluation

Once the population has been generated, either by random initialisation or as a result of a previous generation (lines 1–2, 6, 17 of the Algorithm 1), it is necessary to evaluate each individual. During the fitness evaluation phase, the algorithm calculates the values of two competing objectives for each individual: one to be maximised (Equation (6.1)) and the other to be minimised (Equation (6.2)).

In addition to this analysis, the individuals are classified as feasible or infeasible, based on their conformance to the constraints specific to the problem, such as the ranges of valid harvest ages. Subsequently, feasible solutions are prioritised during the sorting process.

The NSGA-II algorithm then proceeds to sort the combined parent and offspring populations according to the principle of non-dominance, resulting in the formation of multiple Pareto fronts (lines 5, 11–12 in Algorithm 1).

Non-dominated solutions are on the initial front, while subsequent fronts have increasingly dominated solutions. For example, the second front includes one-dominated solution, the third two, and so on. This classification process continues until all individuals have been assigned to a front.

Within each Pareto front, the solutions are further sorted using a crowding distance metric to maintain diversity (line 13 of procedure in Algorithm 1). The crowding distance is a measure of how far a solution is from its neighbours in the objective space. Solutions situated at the boundaries of the search space benefit from higher crowding distances due to the scarcity of their neighbours. By favouring solutions with higher crowding distances, the algorithm ensures that the solution space is explored in a well-distributed manner [97].

6.2.4 Selection

NSGA-II employs an elitist strategy to ensure the preservation of high-quality solutions across generations. The selection process combines the current population with the offspring based on the Binary Tournament Selection operator, as outlined in Deb *et al.* [98]. This operator compares two individuals at a time, first by their Pareto rank and then by their crowding distance if they belong to the same front. The individual with the lowest rank or the largest crowding distance is selected. This process ensures that solutions with superior trade-offs and greater diversity are more likely to be selected, striking an optimal balance between exploration and exploitation (lines 16–18 of the Algorithm 1).

It has been demonstrated that the use of Tournament Selection promotes a more rapid convergence, as described in the framework recommendation [85] (line 11 in the code sample 1). Hence, no alterations were made to this parameter.

6.2.5 Crossover and Mutation

Crossover is a process by which genes from selected individuals are substituted for the next generation, allowing new solutions to be created by merging genetic material from the parent chromosomes. The crossover probability rate determines the likelihood that the chromosomes undergo this process.

It is standard practice to include the mutation as the final operator in an iteration. This operation performs unary transformations (transformations with one operand) on selected individuals. This is achieved by changing alleles on individual chromosomes. In general, the probability of using the mutation operator is fixed throughout all iterations [44]. Typically, all genes are checked and the respective alleles are randomly changed according to a constant low probability [44]. It should be noted that this process has some limitations. Even with the use of library routines to initialise and generate random numbers, the process presents a significant computational challenge.

While GAs have the potential to be an efficient technique, they also have an experimental quality, as observed by Hassanat*et al.* [99]. There is no formula that fits all cases for choosing these operator rates. Generally, a low crossover probability is expected to slow down the convergence process in the first iterations, and too high a probability may lead to saturation around a solution [44].

While Genetic Algorithms can be an effective technique, they also have an experimental quality, as demonstrated by Hassanat *et al.* [99]. It is not possible to apply a single formula to determine the optimal operator rates.

In general, a low crossover probability is expected to slow down the convergence process in the initial iterations. However, a probability that is too high may result in stagnation around a solution.

Therefore, with the mutation operator, lower rates are typically used to avoid converting the evolution programme into a random search approach [100].

According to the work of Hassanat *et al.* [99], for these operators, several studies opt to use ranges for crossover of [0.5–1.0] and for mutation of [0.001–0.5].

In both the custom NSGA-II and the non-modified NSGA-II, the crossover operation used was the framework default. The SBX (Simulated Binary Crossover) method was applied with varying crossover probabilities, as illustrated in code sample 1 lines 12–16.

In terms of the mutation operator, in preliminary tests, it was found that while NSGA-II is efficient and yields a wide range of Pareto-optimal solutions, many of these solutions violate the problem logic, such as ensuring proper forest harvest ages or valid stand allocations. The mutation operator was then adapted to improve the probability of maintaining validity when solutions undergo mutation, as Verma *et al.* [96] suggests.

By integrating problem constraints directly into the mutation process, the operator minimises the generation of invalid solutions. The process probabilistically flips bits for the alleles, while confirming whether the allele randomly chosen to undergo mutation will maintain the chromosome's validity with regard to the harvest-age intervals or species alternatives, lines 17–24 of code 1.

This approach helps to strike a balance between exploring new configurations and maintaining validity, reducing computational waste, and ensuring that the algorithm user only sees the final, valid solutions that optimise both objectives.

6.2.6 Stopping Criteria and Performance Metrics

The evaluation of the quality of the solutions allows for the determination of whether the process should be terminated or continued in order to achieve the optimal results.

The preliminary tests of this module demonstrated that as the number of iterations increased, the number of solutions identified increased exponentially, but the runtime also increased exponentially. In light of the fact that the algorithm module will be available on a Web platform, the number of generations was identified as the primary criterion for termination of the process.

To assess the impact of these modifications, a comparative analysis was carried out using the same scenario and random initial population, contrasting the performance of the standard mutation operator (as required by the original algorithm package) with that of the custom mutation operator. The intention was to assess whether there are significant advantages in using a customised mutation operator to reduce the number of final invalid solutions.

A comprehensive evaluation methodology was used to compare the two approaches within the case study. The performance indicators employed for this comparison included the number of non-dominated solutions, the spacing metric, computational time, and both the evolution and interval values of hypervolume. These metrics were selected because the true Pareto front was not known a priori [96].

The number of non-dominated solutions quantifies the cardinality of the non-dominated set within the final population, representing the solutions located on the approximated Pareto front [101]. An elevated number of non-dominated solutions is indicative of a greater diversity of trade-offs, which is highly desirable in the context of multi-objective optimisation [24].

The spacing metric assesses the uniformity of the distribution of solutions along the approximated Pareto front. A lower spacing value indicates a more uniform distribution of solutions, which in turn provides better coverage of the Pareto front [24, 102].

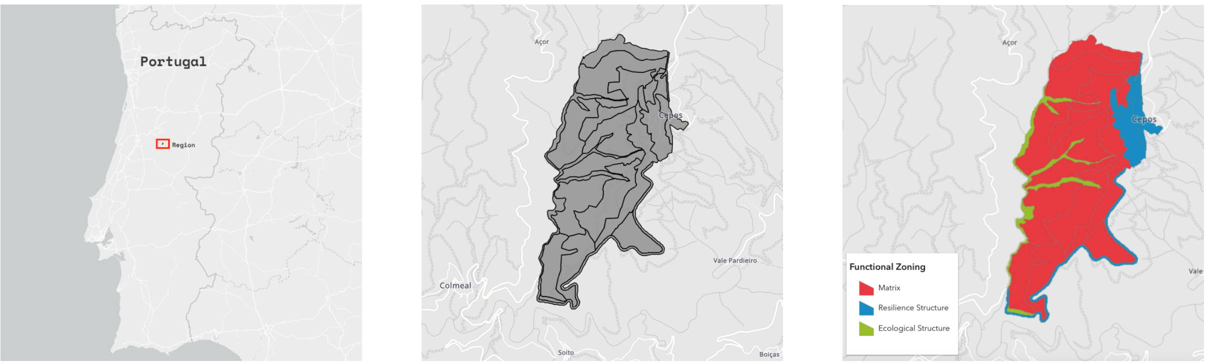
In terms of computational performance, the runtime was employed as a metric to assess the total time taken by the NSGA-II algorithm to reach termination. Although faster algorithms are usually the preferred option, particularly given the intended deployment of this solution on a Web platform, it is important to balance computational efficiency with the quality of the solutions obtained.

The hypervolume metric was used to evaluate the quality of non-dominated solutions [103]. This metric, calculated with a reference point greater than the maximum values of the Pareto front, quantifies the volume of the objective space dominated by the set of solutions [85]. A larger hypervolume indicates a better approximation of the Pareto front [104]. Additionally, the evolution of hypervolume over successive generations provides insights into the algorithm's convergence characteristics, with a higher final hypervolume representing an optimal overall trade-off between the objectives.

The findings of this analysis are presented and discussed in the following sections, outlining the potential advantages of using a customised mutation operator for complex high-dimensional optimisation challenges, such as those encountered in forest management.

6.3 Example Scenario for Evaluating the Algorithm

This study evaluates the performance of the NSGA-II algorithm using a semi-hypothetical dataset for a complex forest scenario. A forest area was selected in the Coimbra District, Portugal (Figure 6.2), considering the physical characteristics of the landscape. The area is divided into three types of landscapes: matrix, resilience, and conservation systems. However, only the matrix landscape type was used for the simulation. Each stand within the forest can be categorised as a pure stand (containing only one species) or a mixed stand (containing more than one species).



(a) Map of Portugal with the study area identified.

(b) Map of the study area used in the simulation.

(c) Map of the study area defined by the three zones.

Figure 6.2: Map of the region selected for this case study.

The forest is divided into 20 stands, 14 of which are part of the landscape matrix and classified as forest stands, making them suitable for the planning problem (Figure 6.2). These 14 stands were selected for testing, reflecting the complexity of a typical real-world problem that is often large in scale. Inventory data, typically provided by forest managers, were collected for these stands. These data include essential information required by the algorithm, such as the total area of the stand and the proportion of usable area.

For existing stands, the collected data included the composition type (pure or mixed), the species present (up to three species per stand), their respective proportions, and biometric parameters to estimate the yield class and the age class for each species. For new stands, the forest manager was asked to estimate up to three future alternative forest compositions and assign a yield class for each species.

To continue the scheduling of harvests, forest managers also determined, for all alternative compositions within each stand and for each species, the minimum and maximum harvest rotation ages and whether to employ a shelterwood or a clear-cut system. Additionally, they established a planning horizon of 100 years, divided into five-year periods.

Table 6.1: Example of the case study stand configuration.

	Minimum Harvest Age	Maximum Harvest Age	Silvicultural System
Stand 1-I			
Pinus pinaster	20	80	Clearcut
Quercus robur	60	120	Shelterwood
Stand 9-I			
Pinus pinaster	30	60	Shelterwood
Quercus robur	40	120	Clearcut
Stand 9-II			
Quercus robur	40	120	Shelterwood
Stand 9-III			
Pinus pinaster	20	50	Shelterwood
		...	

A representation of the context used for the algorithm is provided in Table 6.1. This table outlines the combinations of stands, species, and harvest ages, along with the designated cutting type, as defined for two stands (1 and 9). These examples illustrate two different scenarios: one in which there is an existing forest stand (stand 1) and another in which a new stand will be regenerated (stand 9).

It can be observed that some stands exhibit a single potential combination, implying that the sole combinatorial alteration that may occur within these stands is the modification of the harvest age. However, in stand 9, the algorithm will compare multiple harvest ages for each combination and identify the optimal combination of species and harvest ages that maximises both objectives. In the final solutions for stand 9, only one of the following combinations is permitted: stand 9-I, II, or III. This approach results in a forest management problem consisting of 437 genes per chromosome.

6.4 Algorithm Performance

6.4.1 Setup

All algorithms were implemented and tested on a MacBook Air (Apple M2 CPU, 16 GB RAM) with Apple M2 GPU under the macOS Sonoma (version 14.4.1) operating system. The programming environment used Python 3.11.6, with the NumPy 1.26.3 and SciPy 1.14.0 libraries. These specifications were able to provide sufficient computational power for handling this large-scale task within reasonable time frames.

Table 6.2: Parameters used for testing both NSGA-II algorithms.

	Parameters Tested
Population size	50, 100, 200
Number of generations	100, 200, 500
Crossover probability	0.7, 0.9
Mutation probability	0.5, 0.1, 0.01, 0.002, 0.001

6.4.2 Parametrization

A comparison of the outcomes of the two approaches was achieved by running the basic NSGA-II and the custom mutation operator NSGA-II algorithms 10 times each for each set of parameters, given that it was not possible to maintain consistency in the initial population across all runs.

Table 6.2 outlines the parameters that were modified within the package used for each comparative analysis. The selected crossover probability range was [0.7–0.9], as the values between 0.9 and 0.7 did not result in any significant changes in previous test settings. The mutation rate was established to be between 0.5 and 0.001, as this interval was found to have a significant impact on the number of valid solutions generated. In general, the algorithm displays a preference for maintaining mutation rates near 0. However, in light of the mutation constraints incorporated into the custom NSGA-II, it was deemed prudent to also investigate the 0.5 range, given that previous tests had demonstrated the potential for highly significant solutions to be produced within this interval.

The efficacy of these values has previously been demonstrated, hence their selection for utilisation in the present tests. The number of generations was distributed between 100 and 500. As previously outlined in the initial tests of the algorithm, enhancement was only apparent after 1000 iterations due to the challenge of generating valid solutions. The newly implemented improvements proved to be effective in addressing the previously identified problem. Consequently, it was determined that a 1000-iteration run would not be feasible due to the imposed runtime constraints.

6.4.3 Analysis of the Results

The results for population size (*pop*), number of generations (*n_gen*), crossover probability, and mutation probability that demonstrated optimal performance are presented in Tables 6.3 and 6.4.

Table 6.3: Performance metrics for optimal parameter configuration of the standard NSGA-II.

	Algorithm NSGA-II										
	pop = 50 *n_gen = 100*				*pop = 100* *n_gen = 200*			*pop = 200* *n_gen = 500*			
Crossover	0.9	0.7	0.9		0.7	0.9		0.9	0.7		
Mutation	0.001	0.001	0.5	0.002	0.5	0.1	0.002	0.002	0.001	0.002	0.1
Solutions	**17**	16	15	14	**34**	27	27	**98**	91	89	88
Runtime (s)	2.19	2.41	2.17	2.03	8.5	7.74	8.29	52.54	53.67	51.70	45.41
Spacing	0.08	0.10	0.09	0.11	0.05	0.06	0.05	0.02	0.02	0.02	0.02
Hypervolume	0.70	0.66	0.64	0.58	0.76	0.76	0.71	0.79	0.77	0.77	0.78

Table 6.4: Performance metrics for optimal parameter configuration of the custom NSGA-II.

	Custom NSGA-II											
	pop = 50 *n_gen = 100*				*pop = 100* *n_gen = 200*				*pop = 200* *n_gen = 500*			
Crossover	0.7	0.9		0.7	0.9	0.7		0.9	0.9	0.7		
Mutation	0.5	0.5	0.001	0.002	0.5	0.5	0.1	0.002	0.002	0.05	0.01	0.001
Solutions	**50**	33	21	20	55	**96**	40	36	**123**	112	109	106
Runtime (s)	18.34	7.16	1.94	1.96	26.71	72.92	11.33	7.77	50.80	51.69	129.39	51.72
Spacing	0.02	0.03	0.10	0.10	0.02	0.01	0.04	0.05	0.02	0.02	0.02	0.02
Hypervolume	0.75	0.72	0.70	0.67	0.75	0.76	0.72	0.70	0.78	0.74	0.76	0.78

As illustrated in Table 6.2, a substantial number of configurations were tested, resulting in 72 scenario cases. Therefore, the data presented in the tables mentioned above represent a subset of the complete data. It is important to note that the standard NSGA-II results include a small number of invalid solutions, which are the result of not using the custom mutation operator. Although these instances are not prevalent, they occur.

Despite the variations in the selected case studies, several key conclusions can be drawn from the analysis of the results as a whole, rather than from the small example alone. The customised version of the NSGA-II algorithm consistently demonstrates superior performance compared to the standard version when applied to the specified problem.

The custom Mutation NSGA-II demonstrates a clear advantage in identifying a greater number of solutions. To illustrate, in the *pop = 100/n_gen = 200* parameter configuration, the custom Mutation NSGA-II identifies 55 solutions (*crossover = 0.9, mutation = 0.5*) in 26.71 s, while the standard NSGA-II identifies only 32 solutions under the same parameter settings.

Similarly, in the *pop = 200/n_gen = 500* configuration, the custom mutation NSGA-II produces 123 solutions (*crossover = 0.9, mutation = 0.002*) in 50 s, compared to 98 solutions from the standard NSGA-II algorithm.

The refinement of the mutation operator in the custom NSGA-II effectively balances exploration and exploitation through optimised mutation and crossover rates, aiding in escaping local optima and increasing solution density on the Pareto front.

Crossover and mutation rates have a notable effect on performance: a 0.7 crossover rate generally surpasses the 0.9 rate in the custom NSGA-II. In the *pop = 100/n_gen = 200* configuration, combining a crossover rate of 0.7 with a mutation rate of 0.5 results in 96 solutions, compared to 55 solutions with a crossover rate of 0.9. Lower mutation rates, such as 0.002, produce more solutions when combined with a smaller crossover rate, as they manage to find more solutions closer to the previous solution.

The custom mutation mechanism coupled with a 0.7 crossover rate strikes a superior balance between solution diversity and convergence speed, as indicated by the hypervolume metric. In contrast, standard NSGA-II compensates for its limited adaptability to exploit the solution space with higher crossover rates, such as 0.9.

While both algorithms see gains with larger populations and more generations, the extent varies. The custom NSGA-II almost doubles solution outputs when moving from settings like *pop = 50/n_gen = 100* to *pop = 100/n_gen = 200*. Standard NSGA-II also benefits from scaling, but to a lesser degree than the custom Mutation version.

The custom Mutation NSGA-II has a longer runtime but produces a greater number of solutions. For example, with *pop = 200/n_gen = 500*, it finds 123 solutions in 50 s, whereas the standard NSGA-II achieves 98 solutions in 53 s. Although the standard version can be marginally faster for smaller setups, it offers less diversity in solutions.

The custom NSGA-II's prolonged execution time is justified by its capability to produce a more varied and concentrated Pareto front. This feature is especially beneficial in this problem setting, where diversity and validity of solutions are crucial.

Regarding the spacing metric, the custom Mutation NSGA-II consistently yields lower spacing values, demonstrating better solutions distribution. For example, under the *pop = 200/n_gen = 500* setup, the custom Mutation NSGA-II achieves a spacing of 0.02 across several parameter configurations, whereas the standard NSGA-II also reaches 0.02, but produces fewer solutions overall.

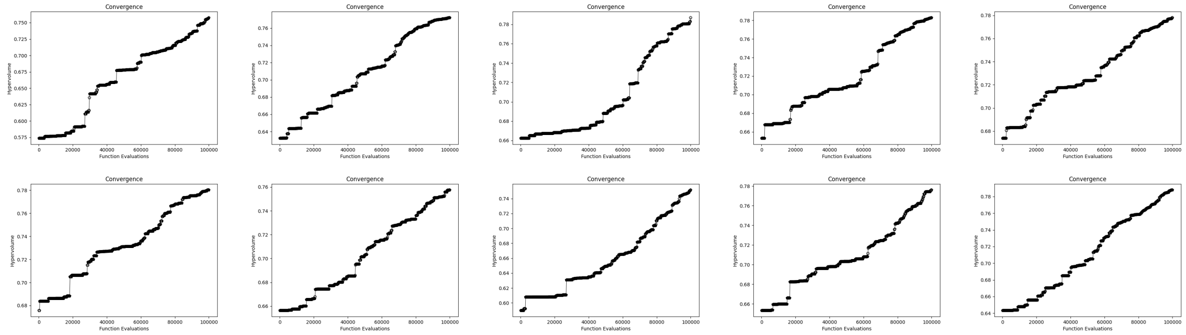


Figure 6.3: Hypervolume convergence curve for standard NSGA-II algorithm.

This indicates that although both algorithms can achieve a uniform distribution, the custom Mutation NSGA-II offers greater density and diversity, thus ensuring a more complete Pareto front.

This analysis can be validated by examining the hypervolume in conjunction with the iterations. For instance, considering the parameter configurations that yield the most solutions for both the custom and standard NSGA-II: *pop = 200/n_gen = 500, crossover = 0.9, mutation = 0.002*. The subsequent plots relate to these configurations, noting that each parameter setup was executed for 10 consecutive runs.

In order to gain an accurate interpretation of the hypervolume metric results, it is crucial to consider not only the minimum and maximum hypervolume indices but also the manner in which the convergence curve progresses over the course of the iterations. Although the final hypervolume values of the algorithms may appear to be similar, an analysis of the convergence behaviour across iterations provides valuable insight into the quality of the solutions.

An examination of the convergence curve for each parameterisation reveals that the customised algorithm displays a more continuous, gradual increase in hypervolume across the ten runs. This indicates a more consistent enhancement in both the convergence and diversity of solutions compared to the standard algorithm, which tends to demonstrate declines or stagnation in the hypervolume index. These observations are illustrated in Figures 6.3 and 6.4. The smoother curve in the customised algorithm indicates a more stable and gradual improvement, reflecting a higher overall solution quality.

Figures 6.5a, 6.5b illustrate the Pareto fronts derived from the specified case study, showing the optimal balance between computational expense and the number of solutions identified. As illustrated in Figure 6.5a, the solution space is limited, with timber volumes ranging from approximately 40,000 to 65,000 and a standard deviation of approximately 12,000 to 20,000. The distribution of points is more sparse, indicating that a smaller number of solutions were identified.

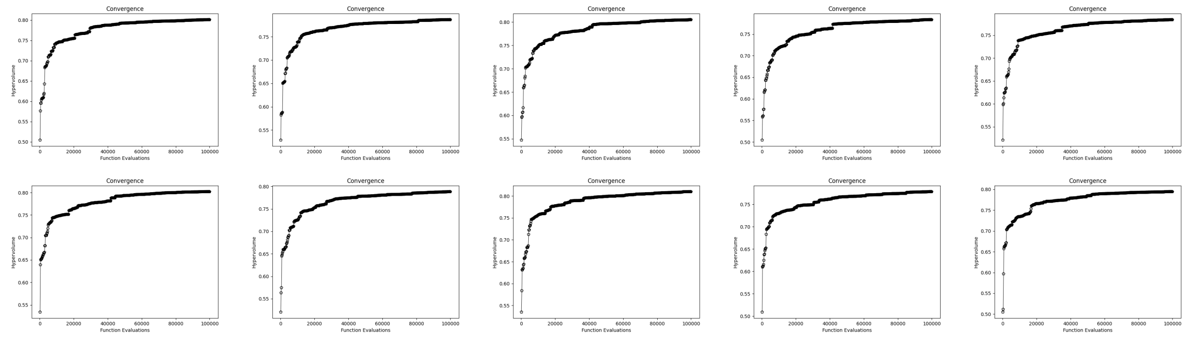
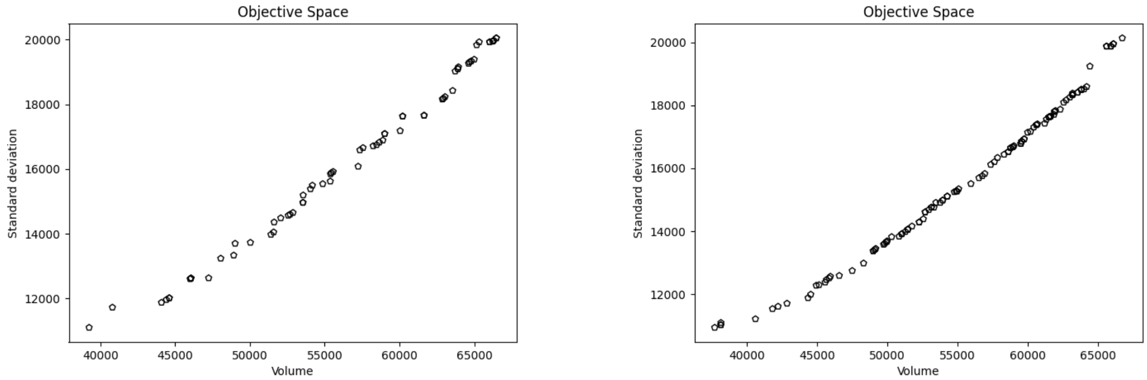


Figure 6.4: Hypervolume convergence curve for custom NSGA-II algorithm.



(a) Pareto front for the NSGA-II.

(b) Pareto front for the custom NSGA-II.

Figure 6.5: Comparison of Pareto fronts for the best case study using NSGA-II and custom NSGA-II algorithms.

Furthermore, the correlation between volume and standard deviation appears to be relatively linear, with the standard deviation gradually increasing as the volume increases. The Pareto front in this case explores a narrower range of higher volumes and higher standard deviations, indicating a lack of diversity in the solutions.

In contrast, Figure 6.5b illustrates a more exhaustive exploration of the solution space. The points in this graph are more densely packed, indicating that a larger set of potential solutions is available. In particular, the customised algorithm appears to provide more trade-offs for lower volumes, as it explores regions with reduced standard deviations.

The goals of maximising timber volume and minimising standard deviation are contradictory. Consequently, each solution on the Pareto front holds equal validity in multi-objective optimisation. In the absence of specific preference information, selecting a particular solution depends entirely on the user's priorities. Each Pareto front point signifies a forest management strategy produced by the algorithm, as shown in Table 6.5.

Table 6.5: Final example of the produced management plans.

Stand	**Species**	**Harvest Age (Year)**	**Volume (m^3)**
1	*Pinus pinaster*	65	3603.22
	Quercus robur	55	4449.66
9	*Pinus pinaster*	50	12,035.93
		...	

6.5 Summary

This chapter introduced a custom NSGA-II algorithm to optimise timber harvest and reduce standard deviation between planning periods. The chapter presented a mathematical formulation of the problem, an overview of the algorithm's modifications, a case study using a Coimbra District dataset, Portugal, and a performance comparison with NSGA-II. The custom algorithm outperformed the standard version, providing more diverse, high-quality solutions. This work offers a practical tool for forest management, designed to achieve a balance between timber production and harvest stability, with the added benefit of integration into a web-based platform. The results of the algorithm have been published, demonstrating the effectiveness of the algorithm in a forest management case study.

7 DISCUSSION

This chapter focusses on examining the project objectives, providing a summary of key elements of the research results, and analysing the implications of these results. This analysis aims to enhance the influence of the study in the field by addressing its limitations and situating the findings within the broader context of existing scientific knowledge, ultimately offering a comprehensive conclusion to the research project.

7.1 Overview

This work contributes to the Floresta Digital project, a collaborative initiative between ISEC and ESAC. The objective of the project is to enhance forest resources management with the aid of digital technologies.

The core functionality of the simulator is based on a multistep form and the application of a customised NSGA-II model. This approach ensures that forest stakeholders can introduce their custom data and personalise their forest stands to generate future management alternatives. The simulator employs real-world data from the Coimbra district in Portugal to create an example study to ensure the algorithm's efficacy. This simulator is able to contribute to the field of sustainable forest management.

7.2 Advantages and Limitations

This project presents several advantages, making it a valuable contribution to digital forest management practices. The integration of a custom NSGA-II algorithm ensures efficient optimisation of harvest strategies by balancing timber production with consistent harvest volumes. The web-based platform improves accessibility, enabling users to explore and analyse diverse forest management scenarios. By incorporating real-world data and problem-specific constraints, the tool demonstrates practical applicability and robustness to address forest challenges.

The initial implementation of the first simulator module has met expectations, fulfilling a key project objective. Its integration into the Floresta Digital website represents a significant step forward in advancing digital forest management.

A notable enhancement of the custom NSGA-II algorithm is the introduction of a specific mutation operator to ensure the adherence to predefined age constraints.

The modified and standard versions of NSGA-II were applied to a forest planning problem, with the aim of maximising the harvest volume of timber and minimising the standard deviation of total production while maintaining a minimum harvest volume.

Performance evaluation, using metrics recommended in the literature, demonstrated that the customised NSGA-II algorithm outperforms the standard version by generating a greater number of valid solutions, thus achieving the primary objective of exploring a broader range of potential management scenarios. However, in some instances, outcomes were statistically similar due to the randomisation of the population. Future work might include using larger and more varied datasets to assess scalability and optimising performance by parallelising the algorithm to decrease website waiting times.

While efforts were made to enhance usability, certain limitations remain. For example, manual data input can be time consuming, especially on mobile devices. Despite various attempts to address this issue, the current solution represents the best achievable outcome within the scope of the project and with the approval of the stakeholders.

Furthermore, while current optimisation goals address critical forest concerns, the simulator could be expanded to incorporate factors such as carbon sequestration, further broadening its environmental and economic applications.

The successful application of heuristic techniques in this project underscores their potential for efficient analysis of forest resources. It also highlights the existing gap in forest management tools accessible to stakeholders, reaffirming the importance of making such innovations available to practitioners.

7.3 Main Contributions

This project, developed in partnership with the Coimbra Agricultural School, has made significant contributions to the field of digital forest management. A key output is the publicly accessible forest management simulator, now available on the Floresta Digital website, `https://www.floresta.digital.esac.pt/simulador-alternativas-gestao`. Example files and step-by-step guides have been provided to help users test the tool, promoting the broader adoption and accessibility of advanced forest management technologies.

The work conducted in this project resulted in the publication of the article "Optimising Forest Management Using Multi-Objective Genetic Algorithms" in the journal *Sustainability MDPI* (December 2024) [25]. This publication highlights the application of the customised NSGA-II algorithm in forest management, contributing to the literature by addressing optimisation problems in forestry.

One of the key contributions of this research is the integration of multi-objective optimisation techniques, specifically NSGA-II, into a user-controlled, web-based system.

This development extends previous applications of the algorithm, which were largely confined to stand-alone or specialised contexts, by making the results public and accessible to all stakeholders.

Although few studies directly explore the integration of optimisation techniques into web-based simulations for forest management, this project distinguishes itself through the development of a customised NSGA-II algorithm tailored to address specific challenges. Unlike previous implementations, which focused primarily on general constraints, this work introduces a custom mutation operator to ensure adherence to predefined age constraints, crucial for sustainable harvesting. This enhancement enables the algorithm to outperform standard NSGA-II in generating a greater number of valid solutions while balancing the two objectives, as demonstrated through performance evaluations.

Moreover, this study marks an important achievement as it is, to the best of current knowledge, the first to create an entirely public and unrestricted simulator for forest management.

8 CONCLUSIONS AND FUTURE WORK

This research addresses the growing need for accessible and advanced tools in forest management, particularly for small-scale, non-industrial private landowners who face complex ecological, economic, and operational challenges. By integrating a customised NSGA-II algorithm into the publicly available Floresta Digital platform, this work bridges the gap between theoretical optimisation methods and practical applications. The resulting simulator empowers stakeholders to explore various forest management scenarios, balancing timber production with long-term goals.

A significant contribution of this study is the development of a custom mutation operator within the NSGA-II algorithm, designed to keep the algorithm in accordance with predefined constraints, such as harvest age restrictions. This innovation resulted in consistently superior performance compared to the standard NSGA-II algorithm across 72 test configurations. For example, in the pop $= 200/$n_gen $= 500$ configuration, the customised version identified 123 solutions in 50 seconds, outperforming the 98 solutions of the standard algorithm in the same runtime. These results highlight the effectiveness of the customised NSGA-II in producing a wider and more diverse set of valid solutions.

The custom algorithm demonstrated its ability to effectively balance exploration and exploitation, with optimal parameter configurations, such as a crossover rate of 0.7 combined with a mutation rate of 0.002, resulting in superior solution density and Pareto-front diversity. Hypervolume metrics further validated these findings, revealing that customised NSGA-II consistently achieved smoother convergence curves and higher solution quality over multiple iterations. The Pareto-fronts for both algorithms show a denser and more diverse solution space when using the customised algorithm, particularly in exploring trade-offs for lower timber volumes and reduced variability.

The accessibility of the simulator through a web-based platform represents a milestone in the democratisation of forest management tools. It offers a practical, user-friendly approach that allows decision-makers to simulate and analyse various strategies without the need for expertise in optimisation algorithms, aligning with regional policies like AIGP. The use of real-world data from central Portugal for the case study demonstrates its potential to support sustainable forest management in complex landscapes.

While the project successfully meets its objectives, opportunities for improvement remain. Expanding the simulator to incorporate additional objectives, such as carbon sequestration and biodiversity preservation, could broaden its impact.

Testing the algorithm with larger, more diverse datasets and implementing parallelisation techniques would further enhance scalability and reduce computational time. Addressing usability challenges, such as simplifying data input processes, would also improve adoption among a wider audience.

In conclusion, this research advances the field of forest management by providing a practical, scalable, and public tool that leverages the power of custom heuristic optimisation. By aligning technological innovation with stakeholder needs and environmental priorities, the project establishes a strong foundation for future developments in sustainable forest management and contributes to a greater understanding of how digital tools can support ecological and economic resilience.

REFERENCES

[1] ICNF, "6.º Inventário Florestal Nacional," Instituto da Conservação da Natureza e das Florestas, Tech. Rep., 2015.

[2] ——, "Portugal Perfil Florestal," Instituto da Conservação da Natureza e das Florestas, Tech. Rep., 2021.

[3] M. Hernández-Morcillo, M. Torralba, T. Baiges, A. Bernasconi, G. Bottaro, S. Brogaard, F. Bussola, E. Díaz-Varela, D. Geneletti, C. M. Grossmann, J. Kister, M. Klingler, L. Loft, M. Lovric, C. Mann, N. Pipart, J. V. Roces-Díaz, S. Sorge, M. Tiebel, L. Tyrväinen, E. Varela, G. Winkel, and T. Plieninger, "Scanning the solutions for the sustainable supply of forest ecosystem services in Europe," *Sustainability Science*, vol. 17, no. 5, pp. 2013–2029, sep 2022, publisher: Springer. [Online]. Available: https://link.springer.com/article/10.1007/s11625-022-01111-4

[4] M. Lazdinis, P. Angelstam, and H. Pülzl, "Towards sustainable forest management in the European Union through polycentric forest governance and an integrated landscape approach," *Landscape Ecology*, vol. 34, no. 7, pp. 1737–1749, jul 2019, publisher: Springer Netherlands. [Online]. Available: https://link.springer.com/article/10.1007/s10980-019-00864-1

[5] P. Angelstam, M. Grodzynskyi, K. Andersson, R. Axelsson, M. Elbakidze, A. Khoroshev, I. Kruhlov, and V. Naumov, "Measurement, collaborative learning and research for sustainable use of ecosystem services: Landscape concepts and Europe as Laboratory," *Ambio*, vol. 42, no. 2, pp. 129–145, mar 2013.

[6] A. P. Fischer, "Forest landscapes as social-ecological systems and implications for management," *Landscape and Urban Planning*, vol. 177, pp. 138–147, sep 2018, publisher: Elsevier.

[7] G. H. Aplet and P. S. Mckinley, "A portfolio approach to managing ecological risks of global change," *Ecosystem Health and Sustainability*, vol. 3, no. 2, feb 2017, publisher: Taylor and Francis Ltd.

[8] S. M. Hagerman and R. Pelai, "Responding to climate change in forest management: two decades of recommendations," *Frontiers in Ecology and the Environment*, vol. 16, no. 10, pp. 579–587, dec 2018, publisher: John Wiley & Sons, Ltd. [Online]. Available: https://onlinelibrary.wiley.com/doi/full/10.1002/fee.1974https://onlinelibrary.wiley.com/doi/abs/10.1002/fee.1974https:

//esajournals.onlinelibrary.wiley.com/doi/10.1002/fee.1974

[9] T. A. Ontl, M. K. Janowiak, C. W. Swanston, J. Daley, S. Handler, M. Cornett, S. Hagenbuch, C. Handrick, L. McCarthy, and N. Patch, “Forest Management for Carbon Sequestration and Climate Adaptation,” *Journal of Forestry*, vol. 118, no. 1, pp. 86–101, jan 2020, publisher: Oxford Academic. [Online]. Available: https://dx.doi.org/10.1093/jofore/fvz062

[10] C. W. Swanston, M. K. Janowiak, L. A. Brandt, P. R. Butler, S. D. Handler, P. D. Shannon, A. D. Lewis, K. Hall, R. T. Fahey, L. Scott, A. Kerber, J. W. Miesbauer, L. Darling, L. Parker, and M. S. Pierre, “Forest Adaptation Resources: climate change tools and approaches for land managers, 2nd edition,” *Gen. Tech. Rep. NRS-GTR-87-2. Newtown Square, PA: U.S. Department of Agriculture, Forest Service, Northern Research Station. 161 p. http://dx.doi.org/10.2737/NRS-GTR-87-2.*, vol. 87-2, pp. 1–161, 2016. [Online]. Available: https://research.fs.usda.gov/treesearch/52760

[11] M. Kolström, M. Lindner, T. Vilén, M. Maroschek, R. Seidl, M. J. Lexer, S. Netherer, A. Kremer, S. Delzon, A. Barbati, M. Marchetti, and P. Corona, “Reviewing the Science and Implementation of Climate Change Adaptation Measures in European Forestry,” *Forests 2011, Vol. 2, Pages 961-982*, vol. 2, no. 4, pp. 961–982, nov 2011, publisher: Molecular Diversity Preservation International. [Online]. Available: https://www.mdpi.com/1999-4907/2/4/961/htmhttps://www.mdpi.com/1999-4907/2/4/961

[12] M. Lindner, M. Schwarz, P. Spathelf, J. de Koning, R. Jandl, I. Viszlai, and M. Vančo, *Adaptation to Climate Change in Sustainable Forest Management in Europe*. FOREST EUROPE Liaison Unit Bratislava, aug 2020.

[13] K. M. Schmitt, T. A. Ontl, S. D. Handler, M. K. Janowiak, L. A. Brandt, P. R. Butler-Leopold, P. D. Shannon, C. L. Peterson, and C. W. Swanston, “Beyond planning tools: Experiential learning in climate adaptation planning and practices,” *Climate*, vol. 9, no. 5, p. 76, may 2021, publisher: MDPI AG. [Online]. Available: https://www.mdpi.com/2225-1154/9/5/76/htmhttps://www.mdpi.com/2225-1154/9/5/76

[14] J. Ananda and G. Herath, “A critical review of multi-criteria decision making methods with special reference to forest management and planning,” *Ecological Economics*, vol. 68, no. 10, pp. 2535–2548, aug 2009.

[15] A. Kaya, P. Bettinger, K. Boston, R. Akbulut, Z. Ucar, J. Siry, K. Merry, and C. Cieszewski, “Optimisation in Forest Management,” *Current Forestry Reports*, vol. 2, no. 1, pp. 1–17, mar 2016, publisher: Springer International Publishing. [Online]. Available: https://link.springer.com/article/10.1007/s40725-016-0027-y

[16] J. Carlsson, G. Lidestav, T. Bjärstig, J. Svensson, and E. M. Nordström, "Opportunities for integrated landscape planning - the broker, the arena, the tool," *Landscape Online*, vol. 55, pp. 1–20, 2017, publisher: Universitaet Bonn.

[17] I. D. M. Ciccarino and M. E. d. S. T. Fernandes, *A bibliometric review of stakeholders' participation in sustainable forest management*. Canadian Science Publishing, mar 2024, vol. 54, iSSN: 12086037 Issue: 3 Publication Title: Canadian Journal of Forest Research.

[18] H. M. Hoganson and D. W. Rose, "A Simulation Approach for Optimal Timber Management Scheduling," *Forest Science*, vol. 30, no. 1, pp. 220–238, mar 1984. [Online]. Available: https://doi.org/10.1093/forestscience/30.1.220

[19] J. Borges and H. Hoganson, "Assessing the impact of management unit design and adjacency constraints on forestwide spatial conditions and timber revenues," *Canadian Journal of Forest Research*, vol. 29, pp. 1764–1774, dec 1999.

[20] E. I. Ducheyne, R. R. D. Wulf, and B. D. Baets, "Single versus multiple objective genetic algorithms for solving the even-flow forest management problem," *Forest Ecology and Management*, vol. 201, no. 2-3, 2004.

[21] D. B. Brahim, M. Kobayashi, M. A. Ali, T. Khatir, and M. E. A. E. Elmeliani, "Metaheuristic Optimization Algorithms: an overview," *HCMCOU Journal of Science – Advances in Computational Structures*, feb 2024, publisher: Ho Chi Minh City Open University.

[22] D. G. Fotakis, E. Sidiropoulos, D. Myronidis, and K. Ioannou, "Spatial genetic algorithm for multi-objective forest planning," *Forest Policy and Economics*, vol. 21, pp. 12–19, aug 2012.

[23] A. Falcão and J. Borges, "Designing an Evolution Program for Solving Integer Forest Management Scheduling Models: An Application in Portugal," *Forest Science -Washington-*, vol. 47, pp. 158–168, may 2001.

[24] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, 2002.

[25] I. Castro, R. Salas-González, B. Fidalgo, J. T. Farinha, and M. Mendes, "Optimising forest management using multi-objective genetic algorithms," *Sustainability*, vol. 16, no. 23, 2024.

[26] A. Kangas, M. Kurttila, T. Hujala, K. Eyvindson, and J. Kangas, *Decision support for forest management*. Springer Cham, nov 2015. [Online]. Available: https://doi.org/10.1007/978-3-319-23522-6

[27] P. Bettinger, K. Boston, J. P. Siry, and D. L. Grebner, *Forest Management and planning*. Academic Press, nov 2016.

[28] M. Mahnken, M. Cailleret, A. Collalti, C. Trotta, C. Biondo, E. D'Andrea, D. Dalmonech, G. Marano, A. Mäkelä, F. Minunno, M. Peltoniemi, V. Trotsiuk, D. Nadal-Sala, S. Sabaté, P. Vallet, R. Aussenac, D. R. Cameron, F. J. Bohn, R. Grote, A. L. D. Augustynczik, R. Yousefpour, N. Huber, H. Bugmann, K. Merganičová, J. Merganic, P. Valent, E. Lasch-Born, and C. P. O. Reyer, "Accuracy, realism and general applicability of european forest models," *Global Change Biology*, vol. 28, no. 23, p. 6921–6943, Dec. 2022.

[29] Y. Shan, "Examining the potential of particle swarm optimization for spatial forest planning and developing a solution quality index for heuristic techniques," PhD Thesis, University of Georgia, 2010.

[30] V. Rieser, D. T. Robinson, D. Murray-Rust, and M. Rounsevell, "A comparison of genetic algorithms and reinforcement learning for optimising sustainable forest management," *GeoComputation*, nov 2011. [Online]. Available: http://www.geocomputation.org/2011/papers/rieser.pdf

[31] L. N. Sotomayor, M. J. Cracknell, and R. Musk, "Supervised machine learning for predicting and interpreting dynamic drivers of plantation forest productivity in northern Tasmania, Australia," *Computers and Electronics in Agriculture*, vol. 209, p. 107804, nov 2023, publisher: Elsevier BV. [Online]. Available: https://doi.org/10.1016/j.compag.2023.107804

[32] FAO, *The State of the World's Forests 2024 – Forest-sector innovations towards a more sustainable future*. Rome, Italy: FAO, Jul. 2024. [Online]. Available: https://doi.org/10.4060/cd1211en

[33] P. Back, "AI for Optimal and Sustainable Forest Management," PhD Thesis, Aalto University, 2022, backup Publisher: School of Business ISBN: 978-952-64-0902-3 (electronic), 978-952-64-0901-6 (printed) ISSN: 1799-4942 (electronic), 1799-4934 (printed), 1799-4934 (ISSN-L). [Online]. Available: http://urn.fi/URN:ISBN:978-952-64-0902-3

[34] E. F. Bell, "Mathematical Programing in Forestry," *Journal of Forestry*, vol. 75, no. 6, 1977.

[35] J. Buongiorno and J. K. Gilless, *Decision Methods for Forest Resource Management*. Academic Press, 2003, iSSN: 0717-6163 Publication Title: Academic Press.

[36] G. B. Dantzig, "Maximization of a linear function of variables subject to linear inequalities," *Activity analysis of production and allocation*, vol. 13, pp. 339–347, 1951.

[37] L. S. Davis, K. N. Johnson, P. Bettinger, and T. E, *Forest Management: To Sustain Ecological, Economic, and Social Values*. Waveland Pr Inc, 2001, publication Title: McGraw-Hill series Forest Resources.

[38] B. Muys, J. Hynynen, M. Palahí, M. J. Lexer, M. Fabrika, H. Pretzsch, F. Gillet, E. Briceño, G.-J. Nabuurs, and V. Kint, "Simulation tools for decision support to adaptive forest management in Europe," *Forest Systems*, vol. 3, no. 4, p. 86, nov 2011, publisher: National Institute for Agricultural and Food Research and Technology. [Online]. Available: https://doi.org/10.5424/fs/201019s-9310

[39] W. White, "A dynamic programming approach to optimal forest stand management regimes in Coastal British Columbia," PhD Thesis, Simon Fraser University, 1988.

[40] T. Heikkinen, "On optimal forest management: a dynamic programming approach," *International Journal of Society Systems Science*, vol. 3, no. 3, p. 217, 2011, publisher: Inderscience Publishers.

[41] Z. Liu, C. Peng, T. Work, J.-N. Candau, A. Desrochers, and D. Kneeshaw, "Application of machine-learning methods in forest ecology: recent progress and future challenges," *Environmental Reviews*, vol. 26, pp. 339–350, jul 2018.

[42] B. R. Sturtevant, A. Fall, D. Kneeshaw, N. P. P. Simon, M. J. Papaik, K. Berninger, F. Doyon, D. G. Morgan, and C. Messier, "A Toolkit Modeling Approach for Sustainable Forest Management Planning: Achieving Balance between Science and Local Needs," *Ecology and Society*, vol. 12, no. 2, nov 2007, publisher: Resilience Alliance. [Online]. Available: https://doi.org/10.5751/es-02102-120207

[43] G. Liu, S. Han, X. Zhao, J. D. Nelson, H. Wang, and W. Wang, "Optimisation algorithms for spatially constrained forest planning," *Ecological Modelling*, vol. 194, no. 4, 2006.

[44] C. Reeves, *Modern Heuristic Techniques for Combinatorial Problems*. John Wiley & Sons, Inc.605 Third Ave. New York, NYUnited States, 1993, publication Title: Department of Statistics and Operational Research.

[45] J. Kangas, "The Analytic Hierachy Process (AHP): Standard Version, Forestry Application and Advances," in *Multiple Use of Forests and Other Natural Resources*. Springer, Dordrecht, 1999, pp. 96–105.

[46] Y. Shan, P. Bettinger, C. Cieszewski, and W. Wang, "Pitfalls and potential of particle swarm optimization for contemporary spatial forest planning," *Forest Systems*, vol. 21, no. 3, 2012.

[47] E. Ducheyne, "Multiple objective forest management using GIS and genetic optimisation techniques," PhD Thesis, Ghent University, 2003. [Online]. Available: http://doi.org/1854/254

[48] T. Pukkala, "Introduction to Multi-Objective Forest Planning," in *Multi-objective Forest Planning*. Springer, Dordrecht, 2002, pp. 1–19, section: 1.

[49] J. H. Holland, *Adaptation in natural and artificial systems*. MIT Press, nov 1992.

[50] C. Reeves, "Genetic Algorithms," in *Handbook of Metaheuristics*. Springer, Boston, MA, nov 2010, vol. 146, pp. 109–139.

[51] M. Srinivas and L. M. Patnaik, "Genetic Algorithms: A Survey," *Computer*, vol. 27, no. 6, 1994.

[52] X. Jin, T. Pukkala, and F. Li, "Fine-tuning heuristic methods for combinatorial optimization in forest planning," *European Journal of Forest Research*, vol. 135, no. 4, 2016.

[53] P. Bettinger, D. Graetz, K. Boston, J. Sessions, and W. Chung, "Eight heuristic planning techniques applied to three increasingly difficult wildlife planning problems," *Silva Fennica*, vol. 36, no. 2, nov 2002. [Online]. Available: https://doi.org/10.14214/sf.545

[54] T. Pukkala and M. Kurttila, "Examining the performance of six heuristic optimisation techniques in different forest planning problems," *Silva Fennica*, vol. 39, no. 1, nov 2005. [Online]. Available: https://doi.org/10.14214/sf.396

[55] C. M. Fonseca and P. J. Fleming, "Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization," *Icga*, vol. 93, no. July, 1993.

[56] J. Horn, N. Nafpliotis, and D. E. Goldberg, "Niched Pareto genetic algorithm for multiobjective optimization," in *IEEE Conference on Evolutionary Computation - Proceedings*, vol. 1, 1994.

[57] K. Deb, D. Datta, C. M. Fonseca, F. G. Lobo, P. A. Condado, and J. Seixas, "Multi-Objective Evolutionary Algorithm for Land-Use Management Problem," *International Journal of Computational Intelligence Research*, vol. 3, no. 4, 2007.

[58] E. Ducheyne, R. Wulf, and B. D. Baets, "A Spatial Approach to Forest-Management Optimization: Linking GIS and Multiple Objective Genetic Algorithms," *International Journal of Geographical Information Science*, vol. 20, pp. 917–928, nov 2006.

[59] E. Næsset, "Estimating timber volume of forest stands using airborne laser scanner data," *Remote Sensing of Environment*, vol. 61, no. 2, pp. 246–253, 1997. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0034425797000412

[60] M. Kurttila, "The spatial structure of forests in the optimization calculations of forest planning — a landscape ecological perspective," *Forest Ecology and Management*, vol. 142, no. 1, pp. 129–142, 2001. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378112700003431

[61] M. Neteler and H. Mitasova, *Open source GIS: a GRASS GIS approach*. Springer

US, nov 2008, vol. 773, publication Title: The International Series in Engineering and Computer Science.

[62] X. He, X. Lei, D. Liu, and Y. Lei, "Developing machine learning models with multiple environmental data to predict stand biomass in natural coniferous-broad leaved mixed forests in Jilin Province of China," *Computers and Electronics in Agriculture*, vol. 212, p. 108162, nov 2023, publisher: Elsevier BV. [Online]. Available: https://doi.org/10.1016/j.compag.2023.108162

[63] M. B. Garzón, R. S. D. Dios, and H. S. Ollero, "Effects of climate change on the distribution of Iberian tree species," *Applied Vegetation Science*, vol. 11, no. 2, pp. 169–178, apr 2008.

[64] R. A. Vaca, D. J. Golicher, and L. Cayuela, "Using climatically based random forests to downscale coarse-grained potential natural vegetation maps in tropical Mexico," *Applied Vegetation Science*, vol. 14, no. 3, pp. 388–401, aug 2011.

[65] S. A. Nitoslawski, K. Wong-Stevens, J. W. N. Steenberg, K. Witherspoon, L. Nesbitt, and C. C. K. v. d. Bosch, "The Digital Forest: Mapping a Decade of Knowledge on Technological Applications for Forest Ecosystems," *Earth's Future*, vol. 9, no. 8, p. e2021EF002123, aug 2021, iSBN: 10.1029/2021 Publisher: John Wiley & Sons, Ltd. [Online]. Available: https://onlinelibrary.wiley.com/doi/full/10.1029/2021EF002123https://onlinelibrary.wiley.com/doi/abs/10.1029/2021EF002123https://agupubs.onlinelibrary.wiley.com/doi/10.1029/2021EF002123

[66] W. Peng and O. K. Sadaghiani, "Enhancement of quality and quantity of woody biomass produced in forests using machine learning algorithms," *Biomass and Bioenergy*, vol. 175, p. 106884, nov 2023, publisher: Elsevier BV. [Online]. Available: https://doi.org/10.1016/j.biombioe.2023.106884

[67] X. Jiang, N. Zhang, J. Huang, P. Zhang, and H. Liu, "Analysis of prediction algorithm for forest land spatial evolution trend in rural planning," *Cluster Computing*, vol. 24, no. 1, 2021.

[68] A. Imada, "A Literature Review: Forest Management with Neural Network and Artificial Intelligence," *Communications in Computer and Information Science*, vol. 440, pp. 9–21, 2014, iSBN: 9783319082004 Publisher: Springer Verlag. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-319-08201-1_3

[69] C. Vasilakos, K. Kalabokidis, J. Hatzopoulos, and I. Matsinos, "Identifying wildland fire ignition factors through sensitivity analysis of a neural network," *Natural Hazards*, vol. 50, no. 1, pp. 125–143, dec 2009, publisher: Springer. [Online]. Available: https://link.springer.com/article/10.1007/s11069-008-9326-3

[70] Y. Safi and A. Bouroumi, "Prediction of forest fires using artificial neural net-

works," *Applied Mathematical Sciences*, vol. 7, no. 5-8, pp. 271–286, 2013, publisher: Hikari Ltd.

[71] G. E. Sakr, I. H. Elhajj, G. Mitri, and U. C. Wejinya, "Artificial intelligence for forest fire prediction," *IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM*, pp. 1311–1316, 2010, iSBN: 9781424480319.

[72] M. S. Stupariu, S. A. Cushman, A. I. Pleşoianu, I. Pătru-Stupariu, and C. Fürst, "Machine learning in landscape ecological analysis: a review of recent approaches," *Landscape Ecology 2021 37:5*, vol. 37, no. 5, pp. 1227–1250, dec 2021, iSBN: 0123456789 Publisher: Springer. [Online]. Available: https://link.springer.com/article/10.1007/s10980-021-01366-9

[73] C. Shi and F. Zhang, "A Forest Fire Susceptibility Modeling Approach Based on Integration Machine Learning Algorithm," *Forests*, vol. 14, no. 7, 2023.

[74] D. L. Schmoldt, "Application of Artificial Intelligence to Risk Analysis for Forested Ecosystems," *Risk Analysis in Forest Management*, pp. 49–74, 2001, iSBN: 978-94-017-2905-5. [Online]. Available: https://link.springer.com/chapter/10.1007/978-94-017-2905-5_3

[75] A. Guimarães, M. Valério, B. Fidalgo, R. Salas-Gonzalez, C. Pereira, and M. Mendes, "Cork oak production estimation using a mask r-cnn," *Energies*, vol. 15, no. 24, 2022.

[76] Jetbrains, "Pycharm: the python ide for data science and web development," 2024. [Online]. Available: https://www.jetbrains.com/pycharm/

[77] GitHub, "Github · build and ship software on a single, collaborative platform," 2024. [Online]. Available: https://github.com/

[78] Atlassian, "Manage your team's projects from anywhere | trello," 2024. [Online]. Available: https://trello.com/

[79] D. S. Foundation, "Django," 2024. [Online]. Available: https://www.djangoproject.com/

[80] ——, "Django overview." [Online]. Available: https://www.djangoproject.com/start/overview/

[81] P. Vadapalli, "Top 12 full stack developer frameworks to master in 2024," Oct. 2024. [Online]. Available: https://www.upgrad.com/blog/full-stack-developer-frameworks-to-learn/

[82] nginx, "nginx," 2024. [Online]. Available: https://nginx.org/

[83] Gunicorn, "Gunicorn - python wsgi http server for unix," 2024. [Online]. Available: https://gunicorn.org/

[84] G. A. W. Product, "Gsap." [Online]. Available: https://gsap.com/about

[85] J. Blank and K. Deb, "Pymoo: Multi-Objective Optimization in Python," *IEEE Access*, vol. 8, pp. 89 497–89 509, 2020, publisher: Institute of Electrical and Electronics Engineers Inc.

[86] R. Garett, J. Chiu, L. Zhang, and S. D. Young, "A literature review: Website design and user engagement," *Online journal of communication and media technologies*, vol. 6, no. 3, p. 1–14, Jul. 2016.

[87] M. Simmons, T. Flannery, and R. Byali, "Maximizing online visibility, the importance of search engine optimization on google," *International Journal of Research Publication and Reviews*, vol. 4, p. 1462–1466, May 2023.

[88] I. Red Hat, "What is ci/cd?" Dec. 2023. [Online]. Available: https://www.redhat.com/en/topics/devops/what-is-ci-cd

[89] G. Inc., "What is a distributed version control system?" 2024. [Online]. Available: https://about.gitlab.com/topics/version-control/benefits-distributed-version-control-system/

[90] M. Christou, "Why wireframes are important in the design process," Jan. 2017. [Online]. Available: https://blog.proto.io/wireframes-design-process/

[91] M. R.W, T. A. R. Jenkins, E. D. Mackie, and E. C. Dick, *Forest Yield: A handbook on forest growth and yield tables for British forestry - Forest Research*. Forestry Commission, 2016. [Online]. Available: https://www.forestresearch.gov.uk/publications/forest-yield-a-handbook-on-forest-growth-and-yield-tables-for-british-forestry/

[92] U. Diéguez-Aranda, A. Rojo-Alboreca, F. Castedo-Dorado, J. González, M. Anta, F. Crecente-Campo, J. M. González-González, C. Pérez Cruzado, R. Rodríguez-Soalleiro, C. López-Sánchez, M. Balboa-Murias, J. Varela, F. Sanchez-Rodriguez, and E. Gómez-García, *Herramientas selvícolas para la gestión forestal sostenible en Galicia*. Xunta de Galicia, jan 2009.

[93] H. A. Santos and L. Martins, *Ferramentas de Apoio à Gestão do Pinhal-Bravo*. Ministério da Agricultura, Pescas e Alimentação, 1986, backup Publisher: Ministério da Agricultura, Pescas e Alimentação.

[94] M. S. Patrício, "Análise da potencialidade produtiva do castanheiro em portugal," PhD Thesis, Universidade Técnica de Lisboa, Instituto Superior de Agronomia, Lisboa, 2006. [Online]. Available: http://hdl.handle.net/10198/3894

[95] N. Zarei, A. Azari, and M. M. Heidari, "Improvement of the performance of NSGA-II and MOPSO algorithms in multi-objective optimization of urban water distribution networks based on modification of decision space," *Applied Water*

Science, vol. 12, no. 6, jun 2022, publisher: Springer Science and Business Media Deutschland GmbH.

[96] S. Verma, M. Pant, and V. Snasel, "A Comprehensive Review on NSGA-II for Multi-Objective Combinatorial Optimization Problems," *IEEE Access*, vol. 9, pp. 57 757–57 791, 2021, publisher: Institute of Electrical and Electronics Engineers Inc.

[97] A. Konak, D. W. Coit, and A. E. Smith, "Multi-objective optimization using genetic algorithms: A tutorial," *Reliability Engineering and System Safety*, vol. 91, no. 9, pp. 992–1007, sep 2006.

[98] K. Deb, K. Sindhya, and T. Okabe, "Self-adaptive simulated binary crossover for real-parameter optimization," in *Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation*. Association for Computing Machinery, 2007, pp. 1187–1194. [Online]. Available: https://doi.org/10.1145/1276958.1277190

[99] A. Hassanat, K. Almohammadi, E. Alkafaween, E. Abunawas, A. Hammouri, and V. B. S. Prasath, "Choosing mutation and crossover ratios for genetic algorithms-a review with a new dynamic approach," *Information (Switzerland)*, vol. 10, no. 12, dec 2019, publisher: MDPI AG.

[100] D. Pham and D. Karaboga, *Intelligent optimisation techniques. Genetic algorithms, tabu search, simulated annealing and neural networks*. Springer London, jan 2000.

[101] L. Jiao, R. Shang, F. Liu, and W. Zhang, "Theoretical basis of natural computation," in *Brain and Nature-Inspired Learning Computation and Recognition*. Elsevier, 2020, pp. 81–95.

[102] A. R. Nuhu, X. Yan, D. Opoku, and A. Homaifar, "A niching framework based on fitness proportionate sharing for multi-objective genetic algorithm (MOGA-FPS)," in *GECCO 2021 Companion - Proceedings of the 2021 Genetic and Evolutionary Computation Conference Companion*. Association for Computing Machinery, Inc, jul 2021, pp. 191–192.

[103] E. Zitzler and L. Thiele, "Multiobjective optimization using evolutionary algorithms — A comparative case study," in *Parallel Problem Solving from Nature — PPSN V*, A. E. Eiben, T. Bäck, M. Schoenauer, and H.-P. Schwefel, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 292–301.

[104] E. Zitzler, D. Brockhoff, and L. Thiele, "The hypervolume indicator revisited: On the design of Pareto-compliant indicators via weighted integration," in *Evolutionary Multi-Criterion Optimization: 4th International Conference, EMO 2007, Matsushima, Japan, March 5-8, 2007. Proceedings 4*. Springer, 2007, pp. 862–876.

APPENDICES

Appendix A - Simulator Wireframes

Figure 1: Wireframe illustrating the option to select stands either from a table or through a map in step two of the simulator

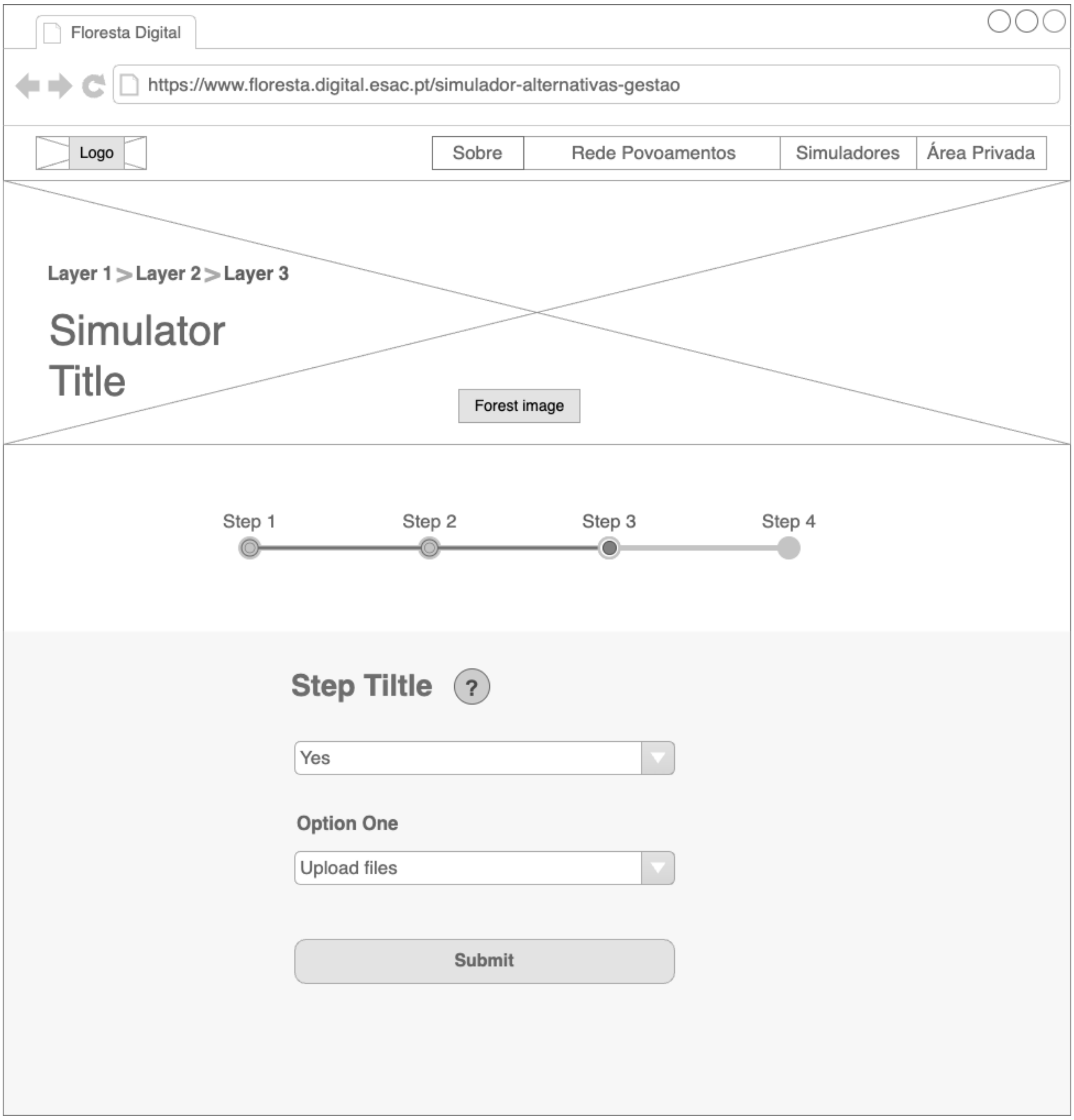


Figure 2: Wireframe illustrating the scenario where two distinct inputs are possible, depending on the option selected for step three of the simulator

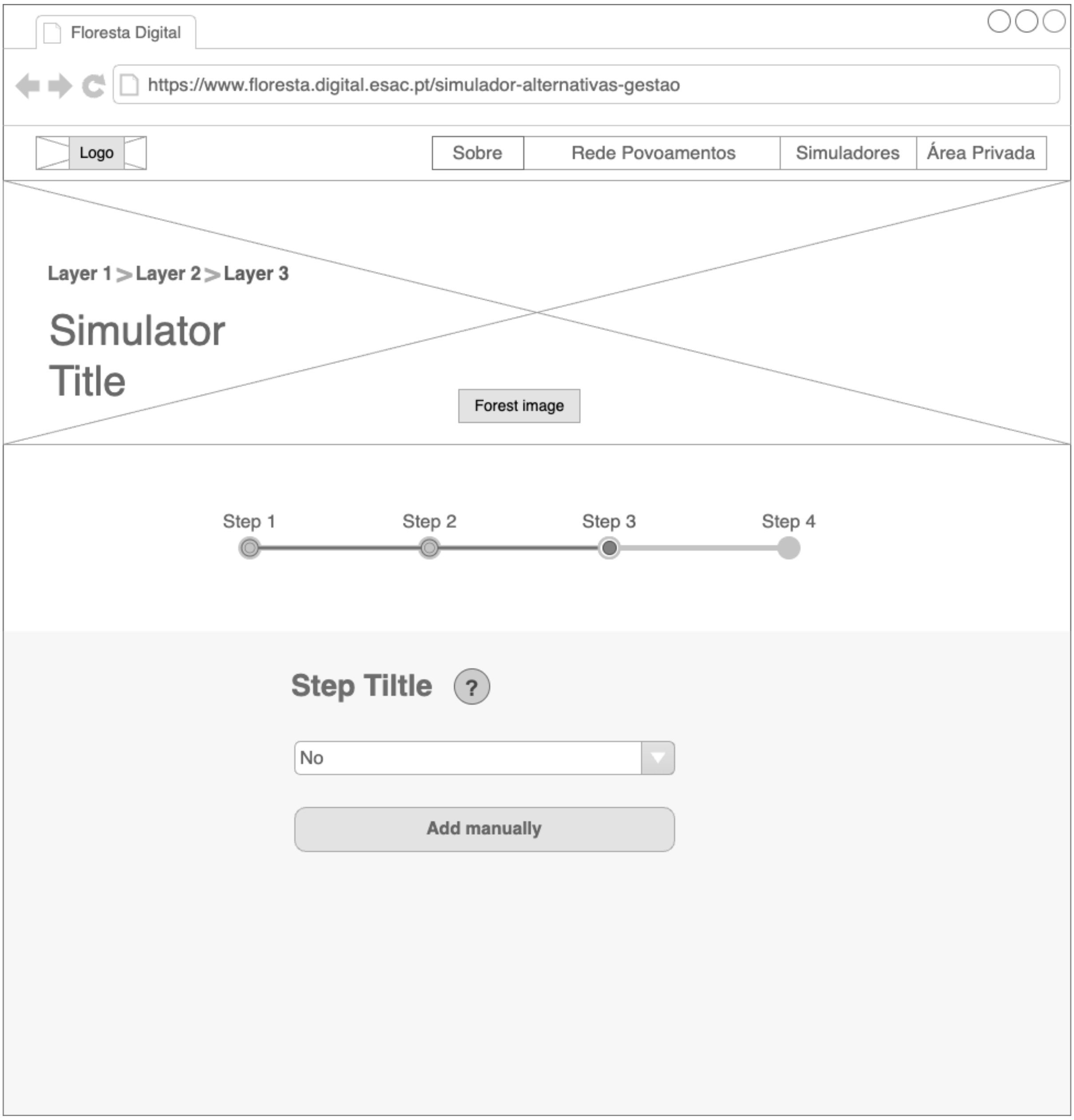
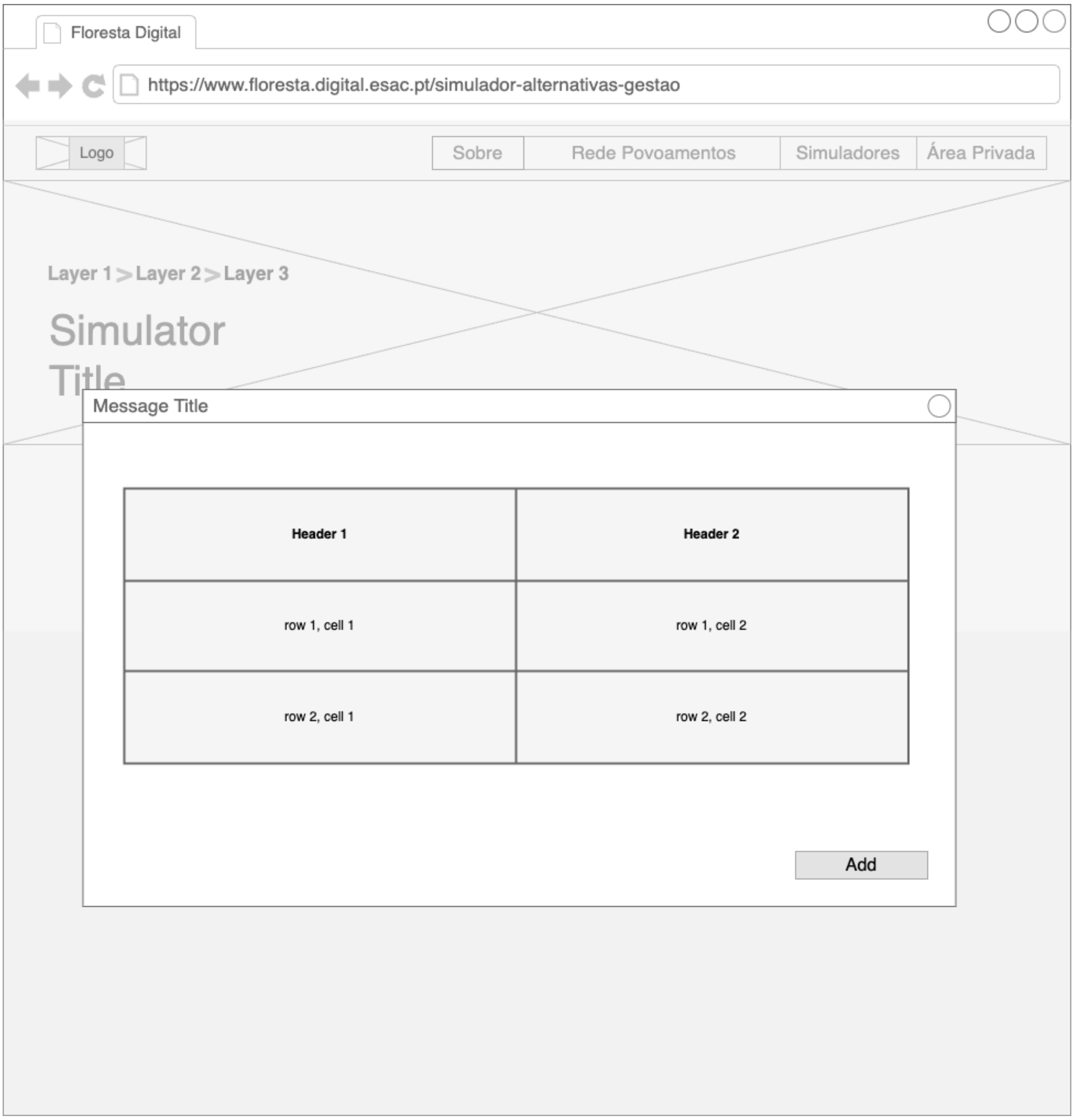


Figure 3: Wireframe for the option to add input manually instead of uploading a file



Header 1	Header 2
row 1, cell 1	row 1, cell 2
row 2, cell 1	row 2, cell 2

Figure 4: Wireframe of a pop-up window where the user can submit data

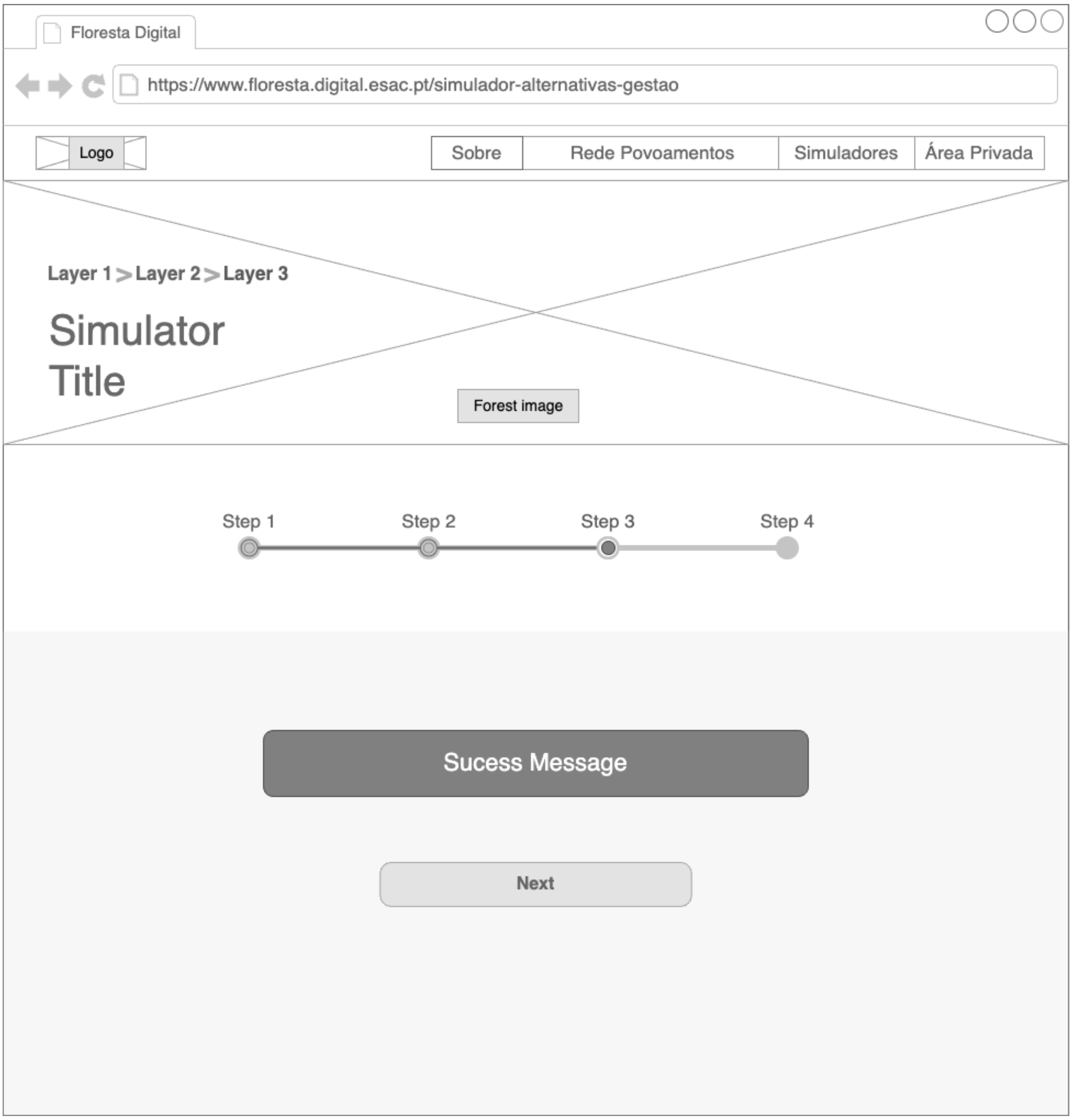


Figure 5: Wireframe for the success message displayed after completing each step of the form

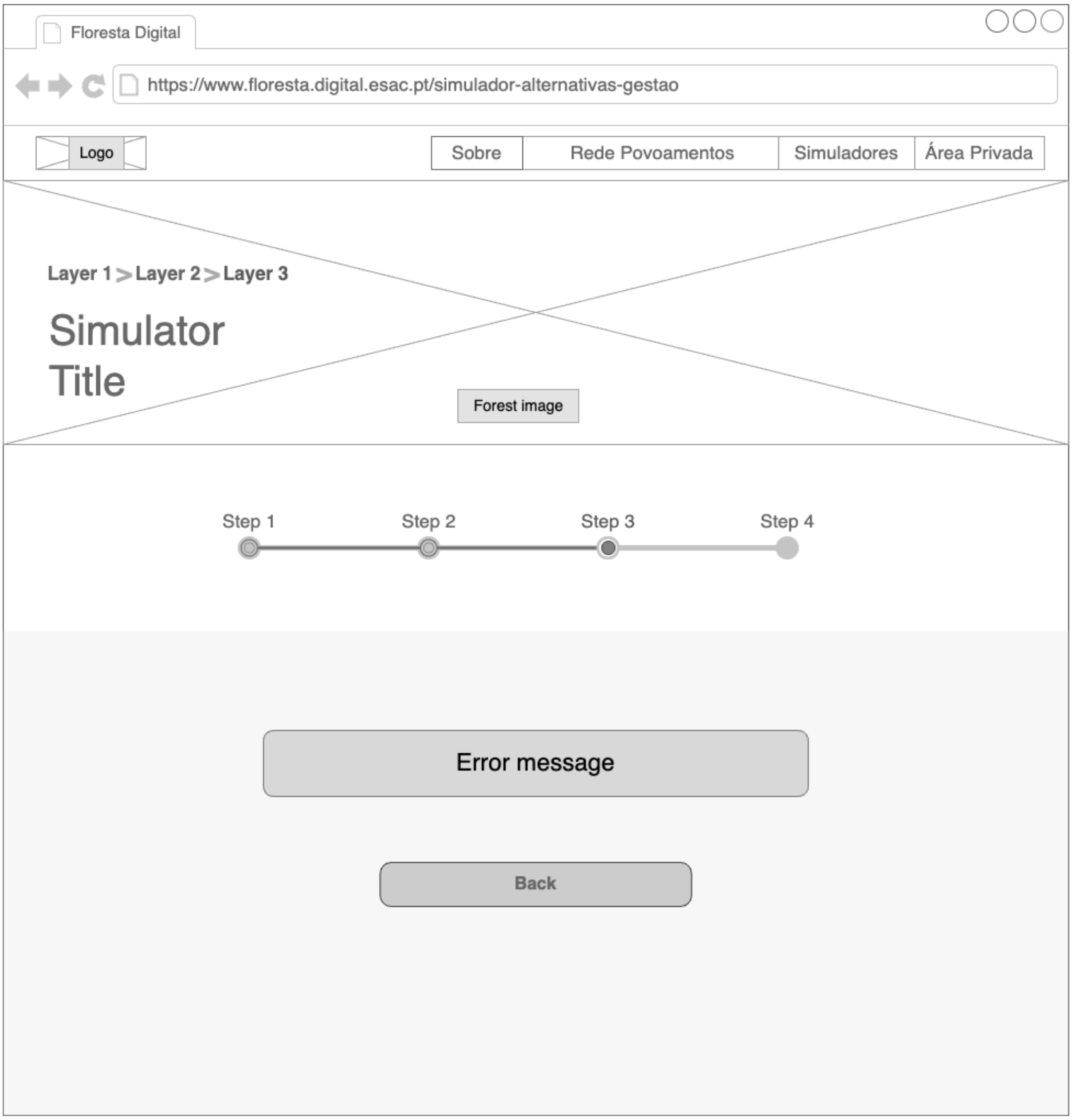


Figure 6: Wireframe for the error message displayed after completing each step of the form

Appendix B - Published Article

sustainability

MDPI

Article

Optimising Forest Management Using Multi-Objective Genetic Algorithms

Isabel Castro [1,2], Raúl Salas-González [2,3], Beatriz Fidalgo [3,*], José Torres Farinha [1,2] and Mateus Mendes [1,2,4,*]

[1] Coimbra Institute of Engineering, Polytechnic University of Coimbra, Rua Pedro Nunes, 3030-199 Coimbra, Portugal; a21280377@isec.pt (I.C.); tfarinha@isec.pt (J.T.F.)
[2] RCM2+, Polytechnic University of Coimbra, Rua Pedro Nunes, 3030-199 Coimbra, Portugal; rsalas@esac.pt
[3] Coimbra Agriculture School, Polytechnic University of Coimbra, Bencanta, 3045-601 Coimbra, Portugal
[4] Institute of Systems and Robotics, Department of Electrical and Computer Engineering, University of Coimbra, 3030-290 Coimbra, Portugal
* Correspondence: bfidalgo@esac.pt (B.F.); mmendes@isec.pt (M.M.)

Abstract: Forest management requires balancing ecological, economic, and social objectives, often involving complex optimisation problems. Traditional mathematical methods struggle with these challenges, leading to the adoption of metaheuristic approaches like the Non-Dominated Sorting Genetic Algorithm II (NSGA-II). This paper introduces a custom NSGA-II algorithm, incorporating a specialised mutation operator to enhance solution generation for multi-objective forest planning. The custom NSGA-II is compared to the standard NSGA-II in a scenario aiming to maximise timber harvest volume and minimise its standard deviation, with a minimum volume constraint. Key performance metrics include non-dominated solutions, spacing, computational cost, and hypervolume. The results demonstrate that the custom NSGA-II provides more valid solutions and better explores the solution space. This approach offers a user-friendly and efficient tool for forest managers, integrating well with Web-based systems for modern, sustainability-oriented forest planning.

Keywords: forest management; optimization; Genetic Algorithm; multi-objective optimization; sustainability; Web integration

check for updates

Citation: Castro, I.; Salas-González, R.; Fidalgo, B.; Farinha, J.T.; Mendes, M. Optimising Forest Management Using Multi-Objective Genetic Algorithms. *Sustainability* **2024**, *16*, 10655. https://doi.org/10.3390/su162310655

Academic Editor: Jozsef Mezei

Received: 17 October 2024
Revised: 24 November 2024
Accepted: 28 November 2024
Published: 5 December 2024

1. Introduction

Forests are not only important for sustainability but in fact indispensable resources for human life on Earth. Hence, forest management has been the subject of much research.

1.1. Context to Forest Management

Forests are complex systems that provide a wide range of ecosystem services, contributing to environmental sustainability and resilience. However, the quantity and diversity of these services depend heavily on the type of forest management applied and the initial conditions of the landscape.

In Portugal, particularly in the central region, the forest landscapes are mainly composed of planted forests owned by many small non-industrial private landowners, whose primary objective is wood production [1,2].

The extensive use of maritime pine and eucalyptus, along with agricultural abandonment and lack of active forest management, has led to large areas of forest monocultures and scrublands. These conditions, combined with orographic and climatic factors typical of the region, as well as climate change, have resulted in widespread forest fires and the spread of invasive woody species. This has caused the loss of ecological and economic value of forest areas and their degradation, significantly altering the landscape and compromising sustainability.

In response to this concern, Portugal has adopted the recommendation of several international bodies and EU policymakers to use a landscape approach to forest management [3–5]. This approach aims to better tailor policy measures and forest management to meet the expectations of local stakeholders and communities and to promote the resilience of forested landscapes [5,6].

A new policy instrument known as AIGP (Integrated Landscape Management Areas) has emerged to promote forest management at the landscape level, proposing a multi-functional model for forest spatial planning. This model is based on functional zoning, which divides the territory into three categories: (1) the resilience structure, which aims to reduce fire hazards; (2) the ecological structure, to ensure the environmental services of the landscape and the conservation of natural areas and biodiversity; and (3) the remaining area, that does not have any designation but will be called a matrix in this work, where wood production and other forest products with economic value can be pursued to meet the needs of forest landowners.

The first two structures are designed to adhere to the spatial environmental restrictions specified in the hierarchical planning system. In contrast, in the matrix, the use of a wide range of forest management models and silvicultural options is encouraged to promote landscape heterogeneity and structural diversity.

It should be noted that this model shares many similarities with those used in other geographical contexts, such as the TRIAD approach, which aims to minimise negative environmental impacts while maintaining the capacity to produce wood and generate economic income [7]. The TRIAD approach proposes creating three zones: one dedicated to nature conservation, another to intensive forestry, and a third where more extensive and diverse forestry practices are applied. Like the AIGP model, the TRIAD approach adopts a landscape-orientated perspective and explicit spatial planning, encouraging the adoption of multiple forest management types across the landscape in a portfolio of management strategies [8] to cope with unexpected results brought by climate change [9,10].

Diversifying management strategies can improve the ecological and economic resilience of the landscape, helping to reduce the likelihood of catastrophic events and the economic dependence on a minimal range of forest products [10].

Management strategies should integrate silvicultural practices to enhance forest resilience to climate change. These practices, among other adaptation measures, include species diversification, the establishment of mixed stands, the promotion of natural regeneration, adjustments to harvesting ages, and the implementation of shelterwood and selective cuts [11–15]. The diversification of silvicultural practices makes the process of selecting management alternatives significantly more complex, requiring careful evaluation of ecological, economic, and operational trade-offs.

Furthermore, historical, political, and social factors significantly influence the effectiveness of forest adaptation strategies, often limiting the universal applicability of findings from global studies in addressing global change [16]. Therefore, it is essential to develop locally tailored adaptation measures that account for specific regional contexts and stakeholder needs, and the long-term sustainability of forest ecosystems.

This adds another layer of complexity to the already intricate challenge of designing forest management compositions and drafting management plans [17,18]. Many authors emphasise the importance of the learning process involved, highlighting its fundamental contribution to the quality and feasibility of the solutions developed [19,20].

The specific social context in Portugal amplifies the importance of the learning process in forest management. Exploring new approaches to adaptive silviculture for forest stands and devising comprehensive management plans for areas with multiple forest owners are indispensable. The inherent complexity of the situation makes it difficult for forest owners and decision-makers to clearly define their preferences without understanding how different management alternatives align with their objectives and potential outcomes.

1.2. Managing Forest Stands Using Genetic Algorithms

Traditional optimisation techniques, such as linear and dynamic programming, have long been applied to multi-objective problems such as harvest scheduling in forest management, with studies by Hoganson et al. [21] and Borges et al. [22] demonstrating their effectiveness in simpler scenarios. However, these methods often struggle to handle larger, more complex problems, where balancing multiple, often conflicting objectives, such as maximising timber yield while minimising environmental impact, becomes computationally prohibitive [23].

In order to address these limitations, heuristic approaches have been employed as a practical alternative, offering near-optimal solutions with manageable computational costs [23,24]. Among these, Genetic Algorithms (GAs) have emerged as a powerful and widely used tool to solve complex, large-scale, multi-objective optimisation problems [25]. GAs utilise evolutionary principles to improve the populations of potential solutions over multiple generations [24]. In contrast to single-objective methods, which consolidate various objectives into a single metric [26], GAs generate a set of solutions that independently balance multiple objectives. As a result, this approach is well suited to scenarios where the optimal solution may be represented by more than one alternative. Thus, in standard multi-objective optimisation, all objectives are distinct measurable criteria, each expressing some aspect requiring optimisation. Assuming all important criteria have been included as objectives, one may be unsure about their relative importance to a decision-maker (and/or about how to normalize them), but one can be certain that the 'ideal' solution will be Pareto-optimal.

A key development in this field is the Non-Dominated Sorting Genetic Algorithm II (NSGA-II), which was introduced by [27]. The effectiveness of NSGA-II in producing a range of Pareto-optimal solutions has been demonstrated, allowing decision-makers to examine a variety of trade-offs rather than a single, potentially biased solution. The core mechanisms of the algorithm, namely, non-dominated sorting and crowding distance, assist in the preservation of diversity within the solution set, thus ensuring a broad selection of high-quality outcomes [27].

This work is based on the underlying principles of NSGA-II and introduces a decision support tool for forest managers. This tool allows for the evaluation and optimisation of management strategies prior to the implementation of a specific plan. The tool has been designed with usability in mind and is accessible via a web-based platform, consisting of two integrated modules. The initial module simulates potential management options for each forest stand, allowing managers to evaluate prospective actions and their consequences for a variety of objectives. The second module employs a modified version of NSGA-II, referred to as custom NSGA-II, which incorporates a specialised mutation operator designed to enhance the generation of diverse solutions tailored to the specific challenges of forest management.

The custom NSGA-II algorithm is evaluated and compared with the standard NSGA-II by applying both methods to a practical example involving forest management. This evaluation is performed with a range of different parameter settings to assess their performance. The forest planning problem under consideration involves two primary goals that the algorithm aims to resolve: maximising timber harvest volume and minimising the standard deviation in harvest volumes across the planning horizon for all stands. A critical constraint ensures that all harvest volumes remain valid (i.e., greater than zero).

The analysis considers a range of parameter configurations, enabling a comprehensive evaluation of the efficiency of the algorithms employed and the diversity of the solutions generated. The outcome of this process is a method that enables forest managers to explore various combinations of stands, assess trade-offs, and analyse different harvest ages over time. It provides an efficient and user-friendly way to optimise forest management practices, helping decision-makers make informed decisions.

2. Literature Review

The problem of forest management has been the subject of extensive research, including traditional exact approaches and heuristic techniques.

2.1. Exact Approaches in Forest Management

In recent years, the global market for wood products has surpassed some countries economies. However, the way this wood is being cultivated and cut is still based on rules of thumb from the 1900s. Despite the need to pay more attention to forests and their ecosystems, there is still difficulty in using several methods already studied and proven to help optimise and achieve sustainable management of forests [28]. In addition to impacting a main operation in forest management, timber harvesting, good or bad practices in forests can also impact other parts of the ecosystem such as CO_2 sequestration, water cycles, local climates, and preservation of biodiversity.

E. Bell defines mathematical programming as "a generic term for a set of methods that can be used to optimise an objective in light of a set of constraints imposed on management activities or constraints imposed on the allocation of land to various uses" [29]. These exact techniques, including linear, goal, and mixed-integer programming, can be used to address spatial forest planning issues if computational needs do not become too demanding and are generally successful in resolving simple planning problems [30].

According to Shan et al. [30], there are various forest planning techniques such as goal, integer, linear, mixed-integer, non-linear, and dynamic programming. These exact methods aim to ensure optimal solutions and are typically used to validate heuristic methods or for small-sized problems. Considering all these techniques, the most common optimisation method for forest management planning is linear programming [31–33]. Linear programming is a useful tool to allocate resources efficiently, taking into account various restrictions and maximising or minimising a goal [34]. Kangas et al. [35] suggest that linear programming can be used to address a variety of sustainability issues, as it allows multiple constraints. However, two essential components are necessary for these models to work: all the variables required to create a plan must be quantified and linear.

Another traditional optimisation technique is dynamic programming [36]. This technique is more versatile and takes special consideration of the sequential decision-making process over time. As such, it becomes especially helpful to address and simplify the ongoing and difficult issues that require a long-term plan for forest management at any age [31,37].

Exact conventional methods have proven their merit in optimising timber production and biodiversity conservation. However, while they may have the advantage of being transparent and interpretable, they may struggle to work for complex, non-linear relationships in forest stands, which is where more modern approaches based on heuristic methods can complement and enhance traditional forest management optimisation practices [38].

2.2. Heuristic Techniques

In the field of forest management, professionals are increasingly being tasked with balancing a broader range of resource objectives than ever before. Modern sustainable forest management has evolved from a focus on timber supply to incorporate more integrated land use planning, taking into account social, economic, and ecological considerations [39].

2.2.1. Multi-Objective Problem Solving

Despite significant advances in computing power and mathematical approaches, these techniques are more difficult to adapt given the increasing diversity of forest planning objectives. The main challenge is the complexity of the problem, because the ecological and human components of decision making have complex relationships [39]. Therefore, in response to computational deficiencies, techniques designed to generate good approximate solutions or predictions have been investigated [40]. These techniques may range from heuristic methods to newer approaches based on artificial intelligence.

In forest planning, pursuing optimal solutions is often a challenging and computationally demanding task. However, heuristics, defined as techniques that seek near-optimal solutions at minimal processing cost, provide a pragmatic solution to this problem according to Reeves et al. [41]. Heuristics are particularly good at producing solutions that find a balance between computational economy and efficacy. Heuristics can also offer adaptability in problem formulation, allowing decision makers to align optimisation problems with diverse goals and preferences [42]. Categorised by solution generation strategies, heuristic methods can be centred on point-based approaches such as Tabu Search and Simulated Annealing, as well as population-based methods like Genetic Algorithms and PSO [30,43].

In forestry, usually, the pursuit of multiple goals lacks a universally agreed criterion for measurement [33]. Assessing the degree of difficulty when meeting multiple goals in management plans remains a challenge. However, most forest management problems are cast as multi-objective optimisation problems. For this type of problem, multiple objectives need maximisation or minimisation, accompanied by meeting constraints to yield feasible solutions. The distinction between single- and multi-objective optimisation lies in the multidimensional space formed by objective functions alongside the typical decision variable space [33,44]. When it comes to reaching the problem's goal, heuristic optimisation techniques do not require a particular form for the objective function. Consequently, the design of the objective functions can vary from case to case.

According to [45], there are four basic compositions to formulate the objective function for a multi-objective forest-planning situation:

1. One goal is either minimised or maximised through the objective function and the other goals become constraints.
2. The objective function measures the deviations of several objective variables from their target levels. These levels are given in other equations of the problem formulation.
3. A single objective is included in an objective function that is augmented with a penalty function.
4. A multi-attribute function is developed and used as the objective function.

Although some of these formulations are similar to traditional mathematical programming approaches, they can be solved using heuristic optimisation techniques. However, they require the planner to define weights or bounds for the objectives. In this study, decision makers do not define these parameters in advance [35].

Given the inherent nature of heuristic problems, that often do not allow for exact solutions, researchers commonly evaluate the quality of the solution using statistical measures such as maximum value, minimum value, mean value, standard deviation, and an estimated global optimum solution [30].

2.2.2. Genetic Algorithms

The term "Genetic Algorithm" (GA) was first used by John Holland in 1975 [46] and laid the foundation for a thriving field of study that has since expanded to cover a wider range of applications than the initial GA. The terms "evolutionary computing" and "evolutionary algorithms" (EAs) are now used to refer to significant developments over the past ten years in the field, which have expanded the concept beyond the original GA [47].

Genetic Algorithms are population-based heuristic techniques based on natural selection and natural genetics. For this reason, the terminology of genetics and biology is used for their description [25]. This algorithm and its variations operate with a population of solutions, termed chromosomes, and combines them to generate new generations of solutions. It uses the concepts of selection, crossover, and mutation to simulate the mechanisms of biological evolution. These algorithms have been successfully used in a variety of areas, including music generation, genetic synthesis, strategic planning, and machine learning [48].

For forest planning, this algorithm provides flexible and efficient approaches to try to find optimal solutions. Given the fact that they search through large solution spaces efficiently, they are suitable for solving combinatorial optimisation problems. They can also handle both spatial and non-spatial objective variables, and so they are an efficient way of solving multi-objective optimisation problems [49].

The extensive use of Genetic Algorithms in forest planning is well documented in various studies [30] and can be implemented in several ways. However, common to all implementations is that they operate with several solutions [50]. This means that GAs operate with a population of solutions. Each member of the population is a chromosome, representing a potential solution. Chromosomes can be viewed as linear formations consisting of contiguous segments, with each segment corresponding to a variable within the problem. Each segment contains one or more elements, called genes, that perform a specific function [25]. From a biological point of view, each gene is responsible for expressing a specific characteristic associated with the individual chromosome.

The initial population of solutions is generated in a random manner and is subsequently subjected to simulated genetic evolution. The evolutionary process is carried out via three genetic operators: selection, crossover, and mutation. Selection is an operation designed to give members of a population with greater fitness a greater chance of survival. In a numerical optimisation problem, the fitness of a solution is measured in relation to the objective function value. The selection process results in a population with an overall higher level of fitness. The population is then subjected to the crossover operation. In this process, the genes of two parent chromosomes are combined to create a new chromosome. Finally, mutation involves the random alteration of one or more genes on one or more chromosomes in the population. In an optimisation context, mutation can help avoid regions of local optima [25].

Several studies have produced different results depending on the characteristics of the problem being solved. These characteristics include whether the problem was single-objective or multi-objective, how complex the constraints or objectives were, and whether the problem was integrated with spatial systems. Most of these studies compared the GA against other heuristic techniques, as previously described.

In [51,52], the authors conducted studies comparing different heuristic optimisation techniques. They found that Genetic Algorithms are highly effective and their adaptability and robustness make them a valuable tool for managing forests, especially when dealing with complex spatial issues. Despite the fact that these studies were conducted, they did not investigate the effectiveness of these algorithms when dealing with uncertainty in forest planning.

In the study in [44], the objective was to implement an optimiser that facilitates simultaneous optimisation of multiple objectives without the need to join these objectives in a single function. In a multi-objective setting, an individual possesses multiple objective values, each pertaining to distinct objective functions. Consequently, the selection criterion for the evolutionary algorithm must be redefined to accommodate an objective value vector while retaining the essence of a single fitness value selection. The investigation compared two multiple objective algorithms: the Multiple Objective Genetic Algorithm (MOGA) developed by Fonseca et al. [53] and the Non-Dominated Sorting Algorithm II (NSGA-II) by Deb et al. [27]. The study aimed to evaluate its performance against a random search strategy to gauge the efficacy of genetic operators. The study revealed the effectiveness of both MOGA and NSGA-II in random search strategies. NSGA-II notably approached the Pareto-optimal front (solution where one of the objectives cannot be improved without sacrificing another one) more closely than MOGA, indicating its better performance in this aspect. However, MOGA excelled in discovering extreme solutions, contributing to a more widespread distribution along the Pareto front. Despite these strengths, neither algorithm managed to pinpoint extreme solutions. NSGA-II demonstrated a faster approximation of the front and displayed evenly spaced solutions, but MOGA slightly outperformed NSGA-II when considering the distance between extreme solutions. Nevertheless, NSGA-II exhibited more robustness in approximating the Pareto-optimal front. Hypervolume measures and other analyses consistently highlighted the superiority of NSGA-II in covering a larger portion of the solution space compared to MOGA. In general, NSGA-II showed better performance in addressing forest management issues. In conclusion, the study emphasises the benefits of employing multiple-objective Genetic Algorithms in forest management planning and suggests the integration of GIS (Geographic Information System) and other heuristic methods for future improvements in efficiency.

In order to address the multiple objectives involved in land use management, a multi-objective evolutionary algorithm is the best solution. Multiple EAs, with slight variations in functionality, have been developed and implemented for these problems. The most widely accepted of these are MOGA [53], NPGA [54], NSGA, SPEA, and NSGA-II [27].

In addition, Deb et al. [55] explored how NSGA-II can be used to optimise multiple, often conflicting, objectives within the context of land use management. The Non-Dominated Sorting Genetic Algorithm-II (NSGA-II) was selected as the optimal solution and designated as NSGA-II-LUM (NSGA-II in Land Use Management). NSGA-II was chosen for its successful application to a broad range of problems. The spatial GIS-based algorithm effectively evaluated three key objectives: maximising economic return, maximising carbon sequestration, and minimising soil erosion. These objectives are particularly important in addressing global warming and soil degradation, making NSGA-II-LUM a valuable asset in contemporary environmental research. Despite data limitations, NSGA-II-LUM showed potential to predict long-term global changes. It uncovers complex relationships and offers valuable information on land use management, promising for dynamic applications [55].

Fotakis et al. [25] introduced a spatial operator to enhance the efficiency of Genetic Algorithms in multi-objective spatial forest planning. The study used Constrained NSGA-II as a standard Genetic Algorithm to evaluate this newly proposed methodology. The focus was on a typical harvest scheduling problem, aiming to maximise timber volume while minimising sediment levels, with constraints on minimum timber yield and even flows. The introduced algorithm, referred to as Spatial NSGA-II, exhibited superior performance for both constrained and unconstrained problems. In particular, it not only yielded improved results, but also managed to keep the old forest together as an emergent outcome, despite not being explicitly targeted. The primary objective of this approach was to facilitate forestry decision making by providing a range of optimal management compositions. The proposed modification of Constrained NSGA-II for spatial forest planning integrates a spatial operator applied at a local level as a secondary type of mutation with a small probability. This operator considers the land unit corresponding to the “mutated” gene and its neighbouring members within the studied area’s plan view. This research underscores the potential of Genetic Algorithms, especially the modified Constrained NSGA-II, in addressing complex spatial forest planning problems. The results highlight its effectiveness in optimising multiple objectives while considering spatial constraints, offering promising applications beyond the scope of this study.

According to Ducheyne et al. [56], when looking at the use of Genetic Algorithms to solve a multi-objective optimisation problem in forest management, it is also essential to take into account spatial objectives. Neglecting spatial intricacies during planning may lead to plans that are misleading or impractical. Genetic Algorithms, particularly when integrated with GIS, offer a solution by providing spatially feasible plans during the optimisation process itself. Insights from [57,58] further emphasise the importance of incorporating spatial objectives and constraints in optimisation of forest management. The Genetic Algorithm and GIS were connected through exchange files, allowing them to communicate independently of the platforms, facilitating optimal task distribution. The results of this integration revealed substantial differences in spatial layout and output levels when spatial information was included, although only a fraction of the Pareto front was explored [59]. This highlights the importance of incorporating spatial data within the optimisation process and demonstrates the potential of a Genetic Algorithm–GIS fusion in the development of spatial decision support systems. It also underscores the risk of omitting spatial-dependent objectives during implementation, which could lead to misleading or unfeasible plans, even in a simple problem with two management compositions. Previous research by the same authors demonstrated the efficiency of GAs over single-objective formulations in forest management optimisation. By integrating the spatial analysis capabilities of a GIS within the optimisation process, spatially feasible plans were constructed during the optimisation.

All studies have in common the need to refer to the fact that the performance of these heuristic methods mainly depends on the parameters that guide their search process. Finding optimal parameter values is a crucial step in obtaining high-quality solutions [40].

3. Problem Definition

In this research, we address a problem that involves a forest composed of numerous and diverse stands, each containing up to three different species with varying current and harvest ages. Each stand may potentially support a variety of different compositions, allowing forest planners to introduce new species into the stand if desired.

The combinatorial problem analysed here aims to find the optimal combinations of stands and harvest ages to achieve two objectives simultaneously: maximising the total timber harvest and minimising the standard deviation of the total timber harvest for each period.

The optimisation goals focus on treating each stand with consideration for the collective objectives of the entire forest. The mathematical formulation of the problem may be described as

$$\begin{aligned} &\text{Maximize} \quad V_{total} = \sum_{n=1}^{N} \sum_{p=1}^{P} \sum_{t=1}^{T} V_{npt} \\ &\text{Subject to} \quad V_{npt} \geq 0, \\ &\qquad\qquad\quad \forall\, n = 1, \ldots, N,\ p = 1, \ldots, P,\ t = 1, \ldots, T \end{aligned} \tag{1}$$

$$\begin{aligned} &\text{Minimize} \quad S_{total} = \sqrt{\frac{\sum_{t=1}^{T} (X - \bar{x})^2}{T-1}} \\ &\text{Subject to} \quad \forall\, t = 1, \ldots, T \end{aligned} \tag{2}$$

where

V_{total} = volume harvested during the planning horizon for all stands N;
N = number of stands;
P = number of species;
T = number of harvest periods;
V_{xtc} = timber volume harvested during period t for species x of stand n;
S_{total} = standard deviation value during the planning horizon for all stands N;
X = stand's timber volume for period t;
$\bar{x}$ = mean of the sample for period t.

The timber volumes for each stand n can be obtained using the following formula:

$$V_{npt} = S_{area} \times P_{proportion} \times (v_{stand} + \sum_{i=1}^{n} v_{thinnings}) \tag{3}$$

where

S_{area} = proportion of usable area in stand n;
$P_{proportion}$ = ratio of species p in stand n;
v_{stand} = volume of the stand for period t;
$v_{thinnings}$ = volume obtained from thinning.

This bi-objective optimisation problem is complemented by a constraint where the total volume cannot be zero, as this represents an invalid solution: $V_{total} > 0$. Moreover, in the course of validating the solutions generated by the algorithm, any solution that exhibited harvest age values not within the specified ranges was considered to violate the constraints, according to the first objective function (1).

3.1. Strategy for Multi-Objective Forest Management

The design of an evolutionary strategy to address the multi-objective forest management problem outlined in Equations (1)–(3) involved a comprehensive analysis of the solution fitness, the selection of genetic operators, and the fine-tuning of the parameters.

A forest consists of a set of stands, each occupying a polygonal area delineated on a map. In the case study, the forest corresponds to the landscape matrix, where the user defines on the map the set of stands that make up the forest (see Figure 1). Each stand has a total area and a usable area, which represents the portion of the total area that can be effectively used for tree coverage.

Each stand S_i possesses specific characteristics that influence the structure of the algorithm.

A stand is considered homogeneous with respect to the attributes used to simulate its growth and yield. These attributes include the composition of the stand, that is, the species present (up to three species) and their respective proportions of occupancy, together with the yield class and age class for each species. These values can be obtained from forest inventories conducted for existing stands or estimated by the user for newly established stands. For newly established stands, the user can define up to three alternative compositions.

Figure 1 illustrates a hypothetical forest composed of two stands: Stand 1 and stand 2.

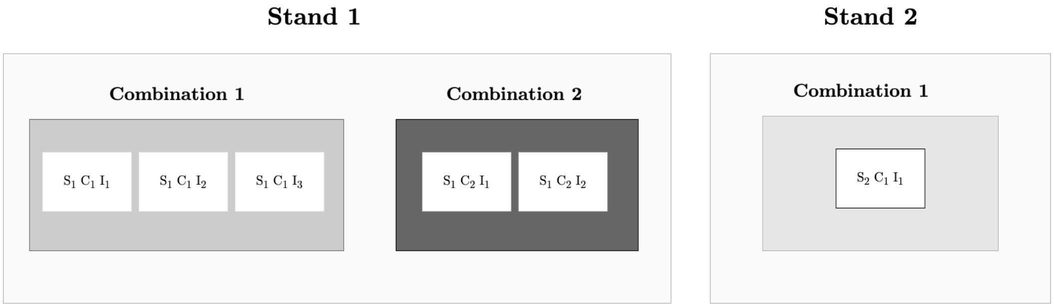


Figure 1. Example of forest stands' possible combinations, where each alternative may have different harvest ages and species occupation.

For stand 1, two alternative compositions are considered, representing two different combinations of species and their respective occupancy proportions. In the first composition, the user specified a mixed stand comprising three species (combination 1), while in the second composition, the user defined a mixed stand with two species (combination 2). In contrast, stand 2 is defined as a pure stand, consisting of only one species.

Each species can be represented by the combination $S_iC_jI_k$, where S_i denotes the stand i (with $1 \leq i \leq m$, where m is the total number of stands), C_j denotes the composition of stand j (with $1 \leq j \leq n$, where n is the maximum number of compositions of the stands), and I_k denotes species k (with $1 \leq k \leq 3$). Here, the variables i, j, and k are indices used to identify specific stands, compositions, and species, respectively.

In addition to the basic parameters characterising the stands, forest management involves a sequence of interventions throughout the life cycle of the stands (referred to as the silvicultural model) to achieve the desired objectives. This model incorporates operations such as regeneration, thinning, the type of harvest cut (clearcut or shelterwood), and harvest age. These interventions are critical because they represent the management options with the greatest influence on both the volume production and the harvest schedule. The user of the Web platform can define these management interventions for each species.

Yield tables are crucial for forest management, as they provide valuable information on tree growth patterns and potential productivity in different forest stands. Forest managers and practitioners use these models to make decisions about the management of individual tree stands or entire estates, forecasting production levels, making commitments to the timber markets, and planning forest operations [60]. Yield classes and volume were estimated using yield tables, which were adjusted to provide volume and thinning estimates at consistent intervals. For each species, three quality classes were considered: high, medium, and low. The yield tables from Diéguez-Aranda et al. [61] were used for *Pseudotsuga menziesii*, *Quercus robur*, and *Betula alba*, while the tables from Santos et al. [62] and Patrício [63] were applied for *Pinus pinaster* and *Castanea sativa*.

Since the yield tables were originally designed for pure stands, their use in mixed stands is limited to mixed stands by patches or groups. Consequently, each species is assigned an independent silvicultural model, which includes defining the minimum and maximum harvest ages that must be considered when generating alternative solutions with the algorithm. It is acknowledged that both harvest age and harvest type can significantly influence the results.

In addition, the problem operates within a specified time frame. The planning horizon can be set to 50, 100, or 200 years, with harvest intervals of either 5 or 10 years.

Preliminary experiments showed enough variable results to allow ongoing improvements to the second project module. At an early stage, modifications were made to enhance the primary structure of the problem, and subsequently adjustments were introduced to optimise the genetic operators. These refinements improved the performance of the algorithm and reduced its computational demands. Reeves et al. [41] highlighted that Genetic Algorithms are naturally adaptable, allowing easy modifications for changes in the original problem.

3.2. Data Structure—Chromosome Representation

The Non-Dominated Sorting Genetic Algorithm (NSGA-II), implemented through the PyMoo package (version 0.6.1.3) in Python [64], was used to find Pareto-optimal solutions, where improving one objective cannot occur without worsening another.

NSGA-II is a state-of-the-art Multi-objective Genetic Algorithm (MOGA) designed to overcome the limitations of classical optimisation methods, such as high computational complexity, non-elitism, and the need to specify a sharing parameter [27]. By integrating elitism, NSGA-II preserves the best solutions from the previous generation while employing genetic operators, mutation, and crossover to generate a new population. This approach not only accelerates convergence towards the optimal solution, but also enhances the overall efficiency of the search process.

This algorithm effectively manages optimisation constraints and maintains diversity in the population by using a crowded comparison operator. The solutions are initially ranked according to dominance, and then sorted according to crowding distance, which contributes to an efficient ranking system that reduces computational complexity [25,65] .

The approach carried out here highlights the efficient use of multi-objective optimisation algorithms to meet specific requirements and goals. Even in the face of strict limits, NSGA-II is able to generate various solutions for a forest management problem.

For the application of NSGA-II, the problem was encoded as follows. The Genetic Algorithm uses a chromosome of length L, where L is determined by the number of compositions, the number of species in each composition, and the defined harvest age periods for each species. Specifically, the length of the chromosome L is given by

$$L = C + \left(\sum_{j=1}^{C} I \right) \times T \tag{4}$$

where C is the number of compositions, I is the number of species in each composition j, and T is the number of harvest periods.

For example, consider the following values:

- Number of compositions: $C = 3$;
- Species per composition:
  - Composition C_1 has $I = 2$;
  - Composition C_2 has $I = 3$;

  - Composition C_3 has $I = 1$.
- Number of harvest periods: $T = 10$.

  Using the formula designated as Equation (4), the following calculation is obtained: Sum of species across all compositions :

$$\sum_{j=1}^{C} I = C_1 + C_2 + C_3 = 2 + 3 + 1 = 6$$

Chromosome length calculation:

$$L = C + 6 \times T = 3 + (6 \times 10) = 3 + 60 = 63$$

As a result, for this test configuration, the length of chromosome L is 63.

Each cell in this chromosome represents a gene, allowing the algorithm to evaluate and optimise based on the encoded information. Although problems typically solved with the PyMoo package are continuous in nature, it is also possible to use other types of variables [64]. In this case, a binary gene type is used. The first set of cells on the chromosome, which represent the stand's alternatives, are subjected to a constraint that allows only one composition per stand to be selected. This constraint is applied across all stands.

Figure 2 illustrates the arrangement of genes within the problem chromosome, as well as the application of genetic operators such as crossover and mutation.

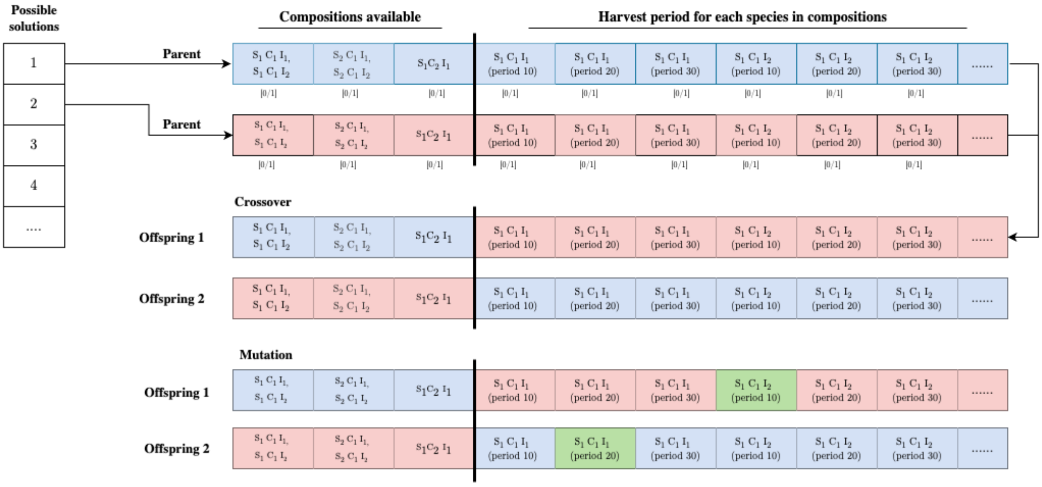


Figure 2. Data encoding for the NSGA-II algorithm.

Each gene in Figure 2 is encoded by the combination $S_iC_jI_k$:

- S_i represents stand i, where $1 \leq i \leq m$, with m being the total number of stands.
- C_j represents stand composition j, where $1 \leq j \leq n$, with n being the maximum number of stand compositions.
- I_k represents species k, where $1 \leq k \leq r$, with r being the maximum number of species.

  In this context

- i, j, and k are index variables used to identify the stands, compositions, and species, respectively.
- m, n, and r are constants denoting the maximum number of stands, compositions, and species in the system.

Subsequently, for each species present in the compositions, the corresponding gene is repeated t times, where T represents the number of periods.

For each group of periods, only one cell can represent the harvest age, ensuring that only one period is designated as the harvest age for that species according to the age range previously defined.

In the course of developing and implementing the NSGA-II algorithm to address this particular problem, it became evident that the scale of the problem posed significant challenges. The complexity of the problem was evidenced by the fact that each chromosome, for a simple forest scenario, consisted of more than 100 genes. This complexity proved detrimental to the algorithm, which was unable to achieve an acceptable ratio of valid to invalid solutions. This issue was particularly notable during the initial stages of the algorithm's execution, with the need for more than 1000 iterations before the algorithm reached an optimal solution.

Therefore, it was established that the algorithm required modification in order to enhance the generation of valid solutions.

This custom NSGA-II algorithm differs from those commonly used in forest management problems, as it accommodates the presence of composition options for each stand [23], harvest-age limits, and a custom initial population and mutation operator. Furthermore, the algorithm is integrated into a novel Web-based platform, facilitating broad accessibility for any user.

The following sections provide a comprehensive analysis of the alterations made to the original NSGA-II algorithm, as outlined by the pseudocode in Algorithm 1.

Algorithm 1 Custom NSGA-II Procedure

```
Require: $N'$, $g$, $f_k(X)$        ▷ $N'$ members evolved over $g$ generations to solve $f_k(X)$
Initialize Population $P_0$
Generate custom population of $N'$ based on problem constraints
Calculate Objectives $V_{total}$ and $S_{total}$
Evaluate Constraint ($V_{total} > 0$)
Assign Rank based on Pareto dominance
Generate Child Population:
    Binary Tournament Selection
    Crossover and custom Mutation
for $i = 1$ to $g$ do
    for each Parent and Child in Population do
        Assign Rank based on Pareto dominance
        Generate sets of non-dominated solutions
        Determine Crowding distance
        Add solutions to next generation starting from the first front until $N'$ individuals
    end for
    Select points on the lower front with high crowding distance
    Create next generation
        Binary Tournament Selection
        Crossover and custom Mutation
end for
Output Results
    Extract and plot best solutions
    Plot convergence and Pareto front
```

3.3. Creation of an Initial Population

A reduction in population size can result in insufficient coverage of the solution space, which may lead to inadequate exploration and an increased risk of premature convergence. On the other hand, an excessively large population can result in a significant increase in computational costs without a corresponding improvement in solution quality [41].

Preliminary tests showed that populations with fewer than 50 individuals exhibited a lack of diversity and frequently converged prematurely, resulting in an insufficient variety of solutions. In contrast, populations of over 500 chromosomes exhibited a considerable increase in computational costs without a corresponding improvement in diversity or the number of solutions, compared to a population size of approximately 200.

To further enhance the performance of the original NSGA-II, a customised operator was developed to generate the initial population (lines 1–2 of the Algorithm 1), replacing the float random sampling provided by the PyMoo framework for NSGA-II [66]. This tailored approach assigns random binary values to each gene to form a chromosome, guaranteeing validity by ensuring that only one harvest age is selected for each species, rather than multiple. This effectively addresses a key issue in the original NSGA-II approach, where a higher proportion of invalid solutions were generated than valid ones.

3.4. Fitness Evaluation

Once the population has been generated, either by random initialisation or as a result of a previous generation (lines 1–2, 6, 17 of the Algorithm 1), it is necessary to evaluate each individual. During the fitness evaluation phase, the algorithm calculates the values of two competing objectives for each individual: one to be maximised (Equation (1)) and the other to be minimised (Equation (2)).

In addition to this analysis, the individuals are classified as feasible or infeasible, based on their conformance to the constraints specific to the problem, such as the ranges of valid harvest ages. Subsequently, feasible solutions are prioritised during the sorting process.

The NSGA-II algorithm then proceeds to sort the combined parent and offspring populations according to the principle of non-dominance, resulting in the formation of multiple Pareto fronts (lines 5, 11–12 in Algorithm 1).

Non-dominated solutions are on the initial front, while subsequent fronts have increasingly dominated solutions. For example, the second front includes one-dominated solution, the third two, and so on. This classification process continues until all individuals have been assigned to a front.

Within each Pareto front, the solutions are further sorted using a crowding distance metric to maintain diversity (line 13 of procedure in Algorithm 1). The crowding distance is a measure of how far a solution is from its neighbours in the objective space. Solutions situated at the boundaries of the search space benefit from higher crowding distances due to the scarcity of their neighbours. By favouring solutions with higher crowding distances, the algorithm ensures that the solution space is explored in a well-distributed manner [67].

3.5. Selection

NSGA-II employs an elitist strategy to ensure the preservation of high-quality solutions across generations. The selection process combines the current population with the offspring based on the Binary Tournament Selection operator, as outlined in Deb et al. [68]. This operator compares two individuals at a time, first by their Pareto rank and then by their crowding distance if they belong to the same front. The individual with the lowest rank or the largest crowding distance is selected. This process ensures that solutions with superior trade-offs and greater diversity are more likely to be selected, striking an optimal balance between exploration and exploitation (lines 16–18 of the Algorithm 1).

It has been demonstrated that the use of Tournament Selection facilitates a more rapid convergence, as described in the framework recommendation [64]. Hence, no alterations were made to this parameter.

3.6. Crossover and Mutation

Crossover is a process by which genes from selected individuals are substituted for the next generation, allowing new solutions to be created by merging genetic material from the parent chromosomes. The crossover probability rate determines the likelihood that the chromosomes undergo this process.

It is standard practice to include mutation as the final operator in an iteration. This operation performs unary transformations (transformations with one operand) on selected individuals. This is achieved by changing alleles in individual chromosomes. In general, the probability of using the mutation operator is fixed throughout all iterations [41]. Typically, all genes are checked and the respective alleles are randomly changed according to a constant low probability [41]. It should be noted that this process has some limitations. Even with the use of library routines for initialising and generating random numbers, the process presents a significant computational challenge.

Although GAs can be a great technique, there is also an experimental quality to them, as [69] shows. There is not one formula that fits all cases for choosing these operator rates. Generally, a low crossover probability is expected to slow down the convergence process in the first iterations, and too high a probability may lead to saturation around a solution [41].

While Genetic Algorithms can be an effective technique, they also have an experimental quality, as demonstrated by Hassanat et al. [69]. It is not possible to apply a single formula to determine the optimal operator rates.

In general, a low crossover probability is expected to slow down the convergence process in the initial iterations. However, a probability that is too high may result in stagnation around a solution. Therefore, with the mutation operator, lower rates are typically used to avoid converting the evolution programme into a random search approach [70].

According to the work of Hassanat et al. [69], for these operators, several studies opt to use ranges for crossover of [0.5–1.0] and for mutation of [0.5–0.001].

In both the custom NSGA-II and the non-modified NSGA-II, the crossover operation used was the framework default. The SBX (simulated binary crossover) method was used with different crossover probabilities.

In terms of the mutation operator, in the preliminary tests it was found that while NSGA-II is efficient and yields a wide range of Pareto-optimal solutions, many of these solutions violate the problem logic, such as ensuring proper forest harvest ages or valid stand allocations. The mutation operator was then adapted to improve the probability of maintaining validity when solutions undergo mutation, as Verma et al. [66] suggests.

By integrating problem constraints directly into the mutation process, the operator minimises the generation of invalid solutions. The process probabilistically flips bits for the alleles, while confirming whether the allele randomly chosen to undergo mutation will maintain the chromosome's validity with regard to the harvest-age intervals or species alternatives.

This approach helps to strike a balance between exploring new configurations and maintaining validity, reducing computational waste and ensuring that the algorithm user only sees the final, valid solutions that optimise both objectives.

3.7. Stopping Criteria and Performance Metrics

The evaluation of the quality of the solutions allows for the determination of whether the process should be terminated or continued in order to achieve the optimal results.

The preliminary tests of this module demonstrated that as the number of iterations increased, the number of solutions identified increased exponentially, but the runtime also increased exponentially. In light of the fact that the algorithm module will be available on a Web platform, the number of generations was identified as the primary criterion for termination of the process.

To assess the impact of these modifications, a comparative analysis was carried out using the same scenario and random initial population, contrasting the performance of the standard mutation operator (as required by the original algorithm package) with that of the custom mutation operator. The intention was to assess whether there are significant advantages in using a customised mutation operator to reduce the number of final invalid solutions.

A comprehensive evaluation methodology was used to compare the two approaches within the case study. The performance indicators employed for this comparison included the number of non-dominated solutions, the spacing metric, computational time, and both the evolution and interval values of hypervolume. These metrics were selected because the true Pareto front was not known a priori [66].

The number of non-dominated solutions quantifies the cardinality of the non-dominated set within the final population, representing the solutions located on the approximated Pareto front [71]. An elevated number of non-dominated solutions is indicative of a greater diversity of trade-offs, which is highly desirable in the context of multi-objective optimisation [27].

The spacing metric assesses the uniformity of the distribution of solutions along the approximated Pareto front. A lower spacing value indicates a more uniform distribution of solutions, which in turn provides better coverage of the Pareto front [27,72].

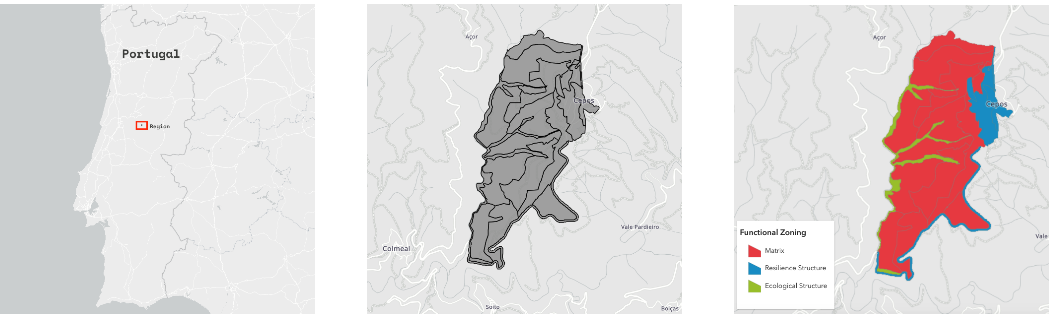
In terms of computational performance, the runtime was employed as a metric to assess the total time taken by the NSGA-II algorithm to reach termination. Although faster algorithms are usually the preferred option, particularly given the intended deployment of this solution on a Web platform, it is important to balance computational efficiency with the quality of the solutions obtained.

The hypervolume metric was used to evaluate the quality of non-dominated solutions [73]. This metric, calculated with a reference point greater than the maximum values of the Pareto front, quantifies the volume of the objective space dominated by the set of solutions [64]. A larger hypervolume indicates a better approximation of the Pareto front [74]. Additionally, the evolution of hypervolume over successive generations provides insights into the algorithm's convergence characteristics, with a higher final hypervolume representing an optimal overall trade-off between the objectives.

The findings of this analysis are presented and discussed in the following sections, outlining the potential advantages of using a customised mutation operator for complex, high-dimensional optimisation challenges, such as those encountered in forest management.

4. Case Study

This study evaluates the performance of the NSGA-II algorithm using a semi-hypothetical dataset for a complex forest scenario. A forest area was selected in the Coimbra District, Portugal (Figure 3), considering the physical characteristics of the landscape. The area is divided into three types of landscapes: matrix, resilience, and conservation systems. However, only the matrix landscape type was used for the simulation. Each stand within the forest can be categorised as a pure stand (containing only one species) or a mixed stand (containing more than one species).



(**a**) (**b**) (**c**)

Figure 3. Map of the region selected for this case study. (**a**) Map of Portugal with the study area identified. (**b**) Map of the study area used in the simulation. (**c**) Map of the study area defined by the three zones.

The forest is divided into 20 stands, 14 of which are part of the landscape matrix and classified as forest stands, making them suitable for the planning problem (Figure 3). These 14 stands were selected for testing, reflecting the complexity of a typical real-world problem that is often large in scale. Inventory data, typically provided by forest managers, were collected for these stands. These data include essential information required by the algorithm, such as the stand's total area and the proportion of usable area. For existing stands, the collected data included the composition type (pure or mixed), the species present (up to three species per stand), their respective proportions, and biometric parameters to estimate the yield class and the age class for each species. For new stands, the forest manager was asked to estimate up to three future alternative forest compositions and provide an expected yield class for each species. To continue the scheduling of harvests, forest managers also determined, for all alternative compositions within each stand and for each species, the minimum and maximum harvest rotation ages and whether to employ a shelterwood or a clearcut system. Additionally, they established a planning horizon of 100 years, divided into five-year periods.

A representation of the context used for the algorithm is provided in Table 1. This table outlines the combinations of stands, species, and harvest ages, along with the designated cutting type, as defined for two stands (1 and 9). These examples illustrate two different scenarios: one in which there is an existing forest stand (stand 1) and another in which a new stand will be regenerated (stand 9).

It can be observed that some stands exhibit a single potential combination, implying that the sole combinatorial alteration that may occur within these stands is the modification of the harvest age. However, in stand 9, the algorithm will compare multiple harvest ages for each combination and identify the optimal combination of species and harvest ages that maximises both objectives. In the final solutions for stand 9, only one of the following combinations is permitted: Stand 9-I, II, or III. This approach results in a forest management problem consisting of 437 genes per chromosome.

Table 1. Example of the case study stand configuration.

	Minimum Harvest Age	**Maximum Harvest Age**	**Silvicultural System**
Stand 1-I			
Pinus pinaster	20	80	Clearcut
Quercus robur	60	120	Shelterwood
Stand 9-I			
Pinus pinaster	30	60	Shelterwood
Quercus robur	40	120	Clearcut
Stand 9-II			
Quercus robur	40	120	Shelterwood
Stand 9-III			
Pinus pinaster	20	50	Shelterwood
		…	

5. Results

All algorithms were implemented and tested on a MacBook Air (Apple M2 CPU, 16 GB RAM) with Apple M2 GPU under the macOS Sonoma (version 14.4.1) operating system. The programming environment used Python 3.11.6, with the NumPy 1.26.3 and SciPy 1.14.0 libraries. These specifications were able to provide sufficient computational power for handling this large-scale task within reasonable time frames.

A comparison of the outcomes of the two approaches was achieved by running the basic NSGA-II and the custom mutation operator NSGA-II algorithms 10 times each for each set of parameters, given that it was not possible to maintain consistency in the initial population across all runs.

Table 2 outlines the parameters that were modified within the package used for each comparative analysis. The crossover probability range selected was [0.7–0.9], as the values between 0.9 and 0.7 did not result in any significant changes in previous test settings. The mutation rate was established to be between 0.5 and 0.001, as this interval was found to have a significant impact on the number of valid solutions generated. In general, the algorithm displays a preference for maintaining mutation rates near 0. However, in light of the mutation constraints incorporated into the custom NSGA-II, it was deemed prudent to also investigate the 0.5 range, given that previous tests had demonstrated the potential for highly significant solutions to be produced within this interval.

The efficacy of these values has previously been demonstrated, hence their selection for utilisation in the present tests. The number of generations was distributed between 100 and 500. As previously outlined in the initial tests of the algorithm, enhancement was only apparent after 1000 iterations due to the challenge of generating valid solutions. The newly implemented improvements proved to be effective in addressing the previously identified problem. Consequently, it was determined that a 1000-iteration run would not be feasible due to the imposed runtime constraints.

Table 2. Parameters used for testing both NSGA-II algorithms.

	Parameters Tested
Population size	50, 100, 200
Number of generations	100, 200, 500
Crossover probability	0.7, 0.9
Mutation probability	0.5, 0.1, 0.01, 0.002, 0.001

The results for population size (*pop*), number of generations *(n_gen)*, crossover probability, and mutation probability that demonstrated optimal performance are presented in Tables 3 and 4.

As illustrated in Table 2, a substantial number of configurations were tested, resulting in 72 scenario cases. Therefore, the data presented in the tables mentioned above represent a subset of the complete data. It is important to note that the standard NSGA-II results include a small number of invalid solutions, which are the result of not using the custom mutation operator. Although these instances are not prevalent, they occur.

Table 3. Performance metrics for optimal parameter configuration of the standard NSGA-II.

	Algorithm NSGA-II											
	pop = 50 *n_gen = 100*				*pop = 100* *n_gen = 200*				*pop = 200* *n_gen = 500*			
Crossover	0.9	0.7	0.9		0.7	0.9			0.9	0.7		
Mutation	0.001	0.001	0.5	0.002	0.5	0.5	0.1	0.002	0.002	0.001	0.002	0.1
Solutions	17	16	15	14	34	32	27	27	98	91	89	88
Runtime (s)	2.19	2.41	2.17	2.03	8.5	9.25	7.74	8.29	52.54	53.67	51.70	45.41
Spacing	0.08	0.10	0.09	0.11	0.05	0.05	0.06	0.05	0.02	0.02	0.02	0.02
Hypervolume	0.70	0.66	0.64	0.58	0.76	0.74	0.76	0.71	0.79	0.77	0.77	0.78

Table 4. Performance metrics for optimal parameter configuration of the custom NSGA-II.

	Custom NSGA-II											
	pop = 50 *n_gen = 100*				*pop = 100* *n_gen = 200*				*pop = 200* *n_gen = 500*			
Crossover	0.7	0.9		0.7	0.9	0.7		0.9	0.9	0.7		
Mutation	0.5	0.5	0.001	0.002	0.5	0.5	0.1	0.002	0.002	0.05	0.01	0.001
Solutions	50	33	21	20	55	96	40	36	123	112	109	106
Runtime (s)	18.34	7.16	1.94	1.96	26.71	72.92	11.33	7.77	50.80	51.69	129.39	51.72
Spacing	0.02	0.03	0.10	0.10	0.02	0.01	0.04	0.05	0.02	0.02	0.02	0.02
Hypervolume	0.75	0.72	0.70	0.67	0.75	0.76	0.72	0.70	0.78	0.74	0.76	0.78

Despite the variations in the selected case studies, several key conclusions can be drawn from the analysis of the results as a whole, rather than from the small example alone.

The customised version of the NSGA-II algorithm consistently demonstrates superior performance compared to the standard version when applied to the specified problem.

The custom Mutation NSGA-II demonstrates a clear advantage in identifying a greater number of solutions. To illustrate, in the *pop = 100/n_gen = 200* parameter configuration, the custom Mutation NSGA-II identifies 55 solutions (*crossover = 0.9, mutation = 0.5*) in 26.71 s, while the standard NSGA-II identifies only 32 solutions under the same parameter settings.

Similarly, in the *pop = 200/n_gen = 500* configuration, the custom mutation NSGA-II produces 123 solutions (*crossover = 0.9, mutation = 0.002*) in 50 s, compared to 98 solutions from the standard NSGA-II algorithm.

The mutation correction in the custom NSGA-II effectively balances exploration and exploitation through optimised mutation and crossover rates, aiding in escaping local optima and increasing solution density on the Pareto front.

Crossover and mutation rates have a notable effect on performance: a 0.7 crossover rate generally surpasses the 0.9 rate in the custom NSGA-II. In the *pop = 100/n_gen = 200* configuration, combining a crossover rate of 0.7 with a mutation rate of 0.5 results in 96 solutions, compared to 55 solutions with a crossover rate of 0.9. Lower mutation rates, such as 0.002, produce more solutions when combined with a smaller crossover rate, as they manage to find more solutions closer to the previous solution.

The custom mutation mechanism coupled with a 0.7 crossover rate strikes a superior balance between solution diversity and convergence speed, as indicated by the hypervolume metric. In contrast, standard NSGA-II compensates for its limited adaptability to exploit the solution space with higher crossover rates, such as 0.9.

While both algorithms see gains with larger populations and more generations, the extent varies. The custom NSGA-II almost doubles solution outputs when moving from settings like *pop = 50/n_gen = 100* to *pop = 100/n_gen = 200*. Standard NSGA-II also benefits from scaling, but to a lesser degree than the custom Mutation version.

The custom Mutation NSGA-II has a longer runtime but produces a greater number of solutions. For example, with *pop = 200/n_gen = 500*, it finds 123 solutions in 50 s, whereas the standard NSGA-II achieves 98 solutions in 53 s. Although the standard version can be marginally faster for smaller setups, it offers less diversity in solutions.

The custom NSGA-II's prolonged execution time is justified by its capability to produce a more varied and concentrated Pareto front. This feature is especially beneficial in this problem setting, where diversity and validity of solutions are crucial.

Regarding the spacing metric, the custom Mutation NSGA-II consistently yields lower spacing values, demonstrating better solutions distribution. For example, under the *pop = 200/n_gen = 500* setup, the custom Mutation NSGA-II achieves a spacing of 0.02 across several parameter configurations, whereas the standard NSGA-II also reaches 0.02, but produces fewer solutions overall. This indicates that although both algorithms can achieve a uniform distribution, the custom Mutation NSGA-II offers greater density and diversity, thus ensuring a more complete Pareto front.

This analysis can be validated by examining the hypervolume in conjunction with the iterations. For instance, considering the parameter configurations that yield the most solutions for both the custom and standard NSGA-II: *pop = 200/n_gen = 500, crossover = 0.9, mutation = 0.002*. The subsequent plots relate to these configurations, noting that each parameter setup was executed for 10 consecutive runs.

In order to gain an accurate interpretation of the hypervolume metric results, it is crucial to consider not only the minimum and maximum hypervolume indices, but also the manner in which the convergence curve progresses over the course of the iterations. Although the final hypervolume values of the algorithms may appear to be similar, an analysis of the convergence behaviour across iterations provides valuable insight into the quality of the solutions.

An examination of the convergence curve for each parameterisation reveals that the customised algorithm displays a more continuous, gradual increase in hypervolume across the ten runs. This indicates a more consistent enhancement in both the convergence and diversity of solutions compared to the standard algorithm, which tends to demonstrate declines or stagnation in the hypervolume index. These observations are illustrated in Figures 4 and 5. The smoother curve in the customised algorithm indicates a more stable and gradual improvement, reflecting a higher overall solution quality.

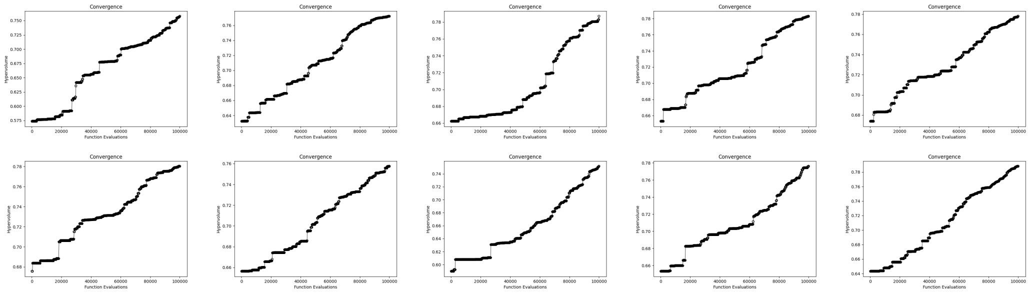


Figure 4. Hypervolume convergence curve for standard NSGA-II algorithm.

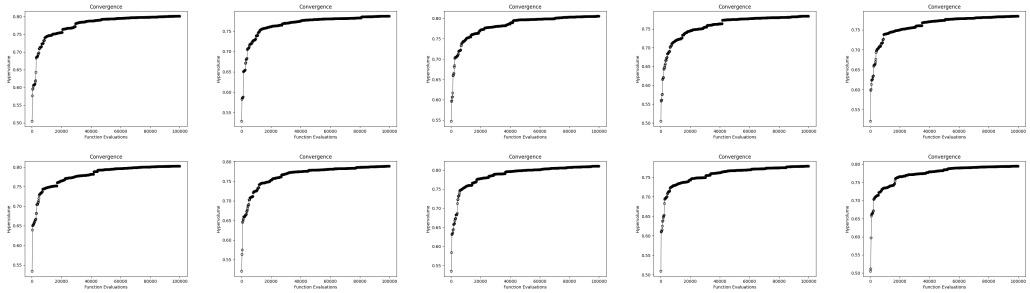
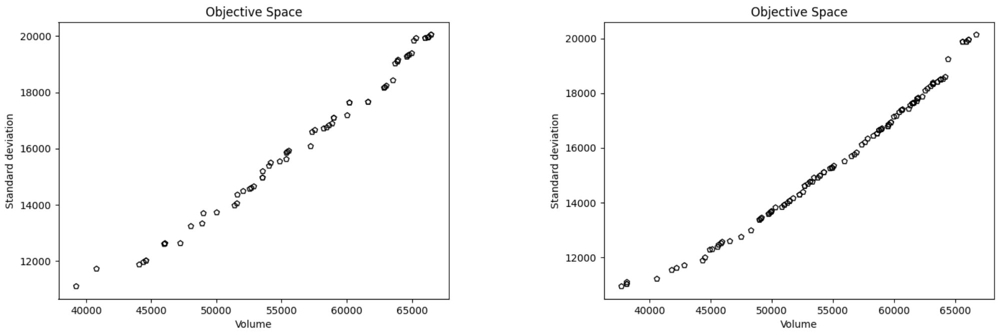


Figure 5. Hypervolume convergence curve for custom NSGA-II algorithm.

Figure 6a,b illustrate the Pareto fronts derived from the specified case study, showing the optimal balance between computational expense and the number of solutions identified. As illustrated in Figure 6a, the solution space is limited, with timber volumes ranging from approximately 40,000 to 65,000 and a standard deviation of approximately 12,000 to 20,000. The distribution of points is more sparse, indicating that a smaller number of solutions were identified. Furthermore, the correlation between volume and standard deviation appears to be relatively linear, with the standard deviation gradually increasing as the volume increases. The Pareto front in this case explores a narrower range of higher volumes and higher standard deviations, indicating a lack of diversity in the solutions.



(**a**) (**b**)

Figure 6. Comparison of Pareto fronts for the best case study using NSGA-II and custom NSGA-II algorithms. (**a**) Pareto front for the NSGA-II. (**b**) Pareto front for the custom NSGA-II.

In contrast, Figure 6b illustrates a more exhaustive exploration of the solution space. The points in this graph are more densely packed, indicating that a larger set of potential solutions is available. In particular, the customised algorithm appears to provide more trade-offs for lower volumes, as it explores regions with reduced standard deviations.

The goals of maximising timber volume and minimising standard deviation are contradictory. Consequently, each solution on the Pareto front holds equal validity in multi-objective optimisation. In the absence of specific preference information, selecting a particular solution depends entirely on the user's priorities. Each Pareto front point signifies a forest management strategy produced by the algorithm, as depicted in Table 5.

Table 5. Final example of the produced management plans.

Stand	Species	Harvest Age (Year)	Volume (m^3)
1	*Pinus pinaster*	65	3603.22
	Quercus robur	55	4449.66
9	*Pinus pinaster*	50	12,035.93
	…		

6. Conclusions

This paper introduces a custom mutation operator tailored to a forest management optimisation scenario. Integrated into the NSGA-II algorithm, the operator ensures that the solutions adhere to the predefined age constraints. The modified and standard NSGA-II were applied to a spatial forest planning problem, aiming to maximise the timber harvest volume and minimisation of the standard deviation in total volume produced, with a constraint on minimum harvest volume.

Performance was evaluated using metrics such as the number of non-dominated solutions, spacing, computational cost, and hypervolume. Both algorithms were run 10 times per parameter configuration and the results averaged. In some cases, the outcomes were statistically similar, likely due to the initialisation of the randomised population.

In general, the customised NSGA-II algorithm is more effective than the standard NSGA-II algorithm in generating a greater number of valid solutions, reflecting the primary objective of exploring the full range of potential solutions in forest management problems. The proposed approach offers a more customised version of NSGA-II, capable of producing valid and diverse compositions for forest management stands in a user-friendly manner, offering forest managers a detailed platform that assesses the possible consequences of their management decisions. Future work includes testing the proposed improved algorithm in larger forest stands.

Author Contributions: Conceptualization, R.S.-G., B.F. and M.M.; methodology, R.S.-G., B.F., J.T.F. and M.M.; software, I.C.; validation, R.S.-G. and B.F.; formal analysis, R.S.-G., B.F., J.T.F. and M.M.; investigation, R.S.-G., B.F., J.T.F. and M.M.; resources, R.S.-G. and B.F.; data curation, I.C.; writing—original draft preparation, I.C.; writing—review and editing, R.S.-G., B.F., J.T.F. and M.M.; supervision, J.T.F. and M.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the European Union—NextGenerationEU, project no. 34 TRANSFORM—Digital transformation of the forestry sector for a more resilient and low-carbon economy. Project co-financed by Centro 2020, Portugal 2020, and the European Union, through the ESF (European Social Fund).

Data Availability Statement: Simulation data available from the corresponding authors upon reasonable request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. ICNF. *6.º Inventário Florestal Nacional*; Technical Report; Instituto da Conservação da Natureza e das Florestas: Algés, Portugal, 2015.
2. ICNF. *Portugal Perfil Florestal*; Technical Report; Instituto da Conservação da Natureza e das Florestas: Algés, Portugal, 2021.
3. Hernández-Morcillo, M.; Torralba, M.; Baiges, T.; Bernasconi, A.; Bottaro, G.; Brogaard, S.; Bussola, F.; Díaz-Varela, E.; Geneletti, D.; Grossmann, C.M.; et al. Scanning the solutions for the sustainable supply of forest ecosystem services in Europe. *Sustain. Sci.* **2022**, *17*, 2013–2029. [CrossRef] [PubMed]

4. Lazdinis, M.; Angelstam, P.; Pülzl, H. Towards sustainable forest management in the European Union through polycentric forest governance and an integrated landscape approach. *Landsc. Ecol.* **2019**, *34*, 1737–1749. [CrossRef]
5. Angelstam, P.; Grodzynskyi, M.; Andersson, K.; Axelsson, R.; Elbakidze, M.; Khoroshev, A.; Kruhlov, I.; Naumov, V. Measurement, collaborative learning and research for sustainable use of ecosystem services: Landscape concepts and Europe as Laboratory. *Ambio* **2013**, *42*, 129–145. [CrossRef] [PubMed]
6. Fischer, A.P. Forest landscapes as social-ecological systems and implications for management. *Landsc. Urban Plan.* **2018**, *177*, 138–147. [CrossRef]
7. Côté, P.; Tittler, R.; Messier, C.; Kneeshaw, D.D.; Fall, A.; Fortin, M.J. Comparing different forest zoning options for landscape-scale management of the boreal forest: Possible benefits of the TRIAD. *For. Ecol. Manag.* **2010**, *259*, 418–427. [CrossRef]
8. Himes, A.; Betts, M.; Messier, C.; Seymour, R. Perspectives: Thirty years of triad forestry, a critical clarification of theory and recommendations for implementation and testing. *For. Ecol. Manag.* **2022**, *510*, 120103. [CrossRef]
9. Himes, A.; Bauhus, J.; Adhikari, S.; Barik, S.K.; Brown, H.; Brunner, A.; Burton, P.J.; Coll, L.; D'Amato, A.W.; Diaci, J.; et al. Forestry in the Face of Global Change: Results of a Global Survey of Professionals. *Curr. For. Rep.* **2023**, *9*, 473–489. [CrossRef]
10. Aplet, G.H.; Mckinley, P.S. A portfolio approach to managing ecological risks of global change. *Ecosyst. Health Sustain.* **2017**, *3*. [CrossRef]
11. Hagerman, S.M.; Pelai, R. Responding to climate change in forest management: Two decades of recommendations. *Front. Ecol. Environ.* **2018**, *16*, 579–587. [CrossRef]
12. Ontl, T.A.; Janowiak, M.K.; Swanston, C.W.; Daley, J.; Handler, S.; Cornett, M.; Hagenbuch, S.; Handrick, C.; McCarthy, L.; Patch, N. Forest Management for Carbon Sequestration and Climate Adaptation. *J. For.* **2020**, *118*, 86–101. [CrossRef]
13. Swanston, C.W.; Janowiak, M.K.; Brandt, L.A.; Butler, P.R.; Handler, S.D.; Shannon, P.D.; Lewis, A.D.; Hall, K.; Fahey, R.T.; Scott, L.; et al. *Forest Adaptation Resources: Climate Change Tools and Approaches for Land Managers*, 2nd ed.; General Technical Reports; U.S. Department of Agriculture, Forest Service, Northern Research Station: Newtown Square, PA, USA, 2016; 161p. [CrossRef]
14. Kolström, M.; Lindner, M.; Vilén, T.; Maroschek, M.; Seidl, R.; Lexer, M.J.; Netherer, S.; Kremer, A.; Delzon, S.; Barbati, A.; et al. Reviewing the Science and Implementation of Climate Change Adaptation Measures in European Forestry. *Forests* **2011**, *2*, 961–982. [CrossRef]
15. Lindner, M.; Schwarz, M.; Spathelf, P.; de Koning, J.; Jandl, R.; Viszlai, I.; Vančo, M. *Adaptation to Climate Change in Sustainable Forest Management in Europe*; Forest Europe Liaison Unit Bratislava: Bonn, Germany, 2020.
16. Schmitt, K.M.; Ontl, T.A.; Handler, S.D.; Janowiak, M.K.; Brandt, L.A.; Butler-Leopold, P.R.; Shannon, P.D.; Peterson, C.L.; Swanston, C.W. Beyond planning tools: Experiential learning in climate adaptation planning and practices. *Climate* **2021**, *9*, 76. [CrossRef]
17. Ananda, J.; Herath, G. A critical review of multi-criteria decision making methods with special reference to forest management and planning. *Ecol. Econ.* **2009**, *68*, 2535–2548. [CrossRef]
18. Kaya, A.; Bettinger, P.; Boston, K.; Akbulut, R.; Ucar, Z.; Siry, J.; Merry, K.; Cieszewski, C. Optimisation in Forest Management. *Curr. For. Rep.* **2016**, *2*, 1–17. [CrossRef]
19. Carlsson, J.; Lidestav, G.; Bjärstig, T.; Svensson, J.; Nordström, E.M. Opportunities for integrated landscape planning—The broker, the arena, the tool. *Landsc. Online* **2017**, *55*, 1–20. [CrossRef]
20. Ciccarino, I.D.M.; Fernandes, M.E.d.S.T. A bibliometric review of stakeholders' participation in sustainable forest management; *Can. J. For. Res.* **2023**, *54*, 252–267. [CrossRef]
21. Hoganson, H.M.; Rose, D.W. A Simulation Approach for Optimal Timber Management Scheduling. *For. Sci.* **1984**, *30*, 220–238. [CrossRef]
22. Borges, J.; Hoganson, H. Assessing the impact of management unit design and adjacency constraints on forestwide spatial conditions and timber revenues. *Can. J. For. Res.* **1999**, *29*, 1764–1774. [CrossRef]
23. Ducheyne, E.I.; Wulf, R.R.D.; Baets, B.D. Single versus multiple objective genetic algorithms for solving the even-flow forest management problem. *For. Ecol. Manag.* **2004**, *201*, 259–273. [CrossRef]
24. Brahim, D.B.; Kobayashi, M.; Ali, M.A.; Khatir, T.; Elmeliani, M.E.A.E. Metaheuristic Optimization Algorithms: An overview. *HCMCOU J. Sci. Adv. Comput. Struct.* **2024**, *14*. [CrossRef]
25. Fotakis, D.G.; Sidiropoulos, E.; Myronidis, D.; Ioannou, K. Spatial genetic algorithm for multi-objective forest planning. *For. Policy Econ.* **2012**, *21*, 12–19. [CrossRef]
26. Falcão, A.; Borges, J. Designing an Evolution Program for Solving Integer Forest Management Scheduling Models: An Application in Portugal. *For. Sci.-Wash.* **2001**, *47*, 158–168. [CrossRef]
27. Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **2002**, *6*, 182–197. [CrossRef]
28. Back, P. AI for Optimal and Sustainable Forest Management. Ph.D. Thesis, Aalto University, Espoo, Finland, 2022.
29. Bell, E.F. Mathematical Programing in Forestry. *J. For.* **1977**, *75*, 317–319. [CrossRef]
30. Shan, Y. Examining the Potential of Particle Swarm Optimization for Spatial Forest Planning and Developing a Solution Quality Index for Heuristic Techniques. Ph.D. Thesis, University of Georgia, Athens, GA, USA, 2010.
31. Buongiorno, J.; Gilless, J.K. *Decision Methods for Forest Resource Management*; Academic Press: Cambridge, MA, USA, 2003.
32. Dantzig, G.B. Maximization of a linear function of variables subject to linear inequalities. *Act. Anal. Prod. Alloc.* **1951**, *13*, 339–347.

33. Davis, L.S.; Johnson, K.N.; Bettinger, P.; Howard, T.E. *Forest Management: To Sustain Ecological, Economic, and Social Values*; Waveland Press Inc.: Long Grove, IL, USA, 2001.
34. Muys, B.; Hynynen, J.; Palahí, M.; Lexer, M.J.; Fabrika, M.; Pretzsch, H.; Gillet, F.; Briceño, E.; Nabuurs, G.J.; Kint, V. Simulation tools for decision support to adaptive forest management in Europe. *For. Syst.* **2011**, *3*, 86. [CrossRef]
35. Kangas, A.; Kurttila, M.; Hujala, T.; Eyvindson, K.; Kangas, J. *Decision Support for Forest Management*; Springer: Cham, Switzerland, 2015. [CrossRef]
36. White, W. A Dynamic Programming Approach to Optimal Forest Stand Management Regimes in Coastal British Columbia. Ph.D. Thesis, Simon Fraser University, Burnaby, BC, Canada, 1988.
37. Heikkinen, T. On optimal forest management: A dynamic programming approach. *Int. J. Soc. Syst. Sci.* **2011**, *3*, 217. [CrossRef]
38. Liu, Z.; Peng, C.; Work, T.; Candau, J.N.; Desrochers, A.; Kneeshaw, D. Application of machine-learning methods in forest ecology: Recent progress and future challenges. *Environ. Rev.* **2018**, *26*, 339–350. [CrossRef]
39. Sturtevant, B.R.; Fall, A.; Kneeshaw, D.; Simon, N.P.P.; Papaik, M.J.; Berninger, K.; Doyon, F.; Morgan, D.G.; Messier, C. A Toolkit Modeling Approach for Sustainable Forest Management Planning: Achieving Balance between Science and Local Needs. *Ecol. Soc.* **2007**, *12*, 7. [CrossRef]
40. Liu, G.; Han, S.; Zhao, X.; Nelson, J.D.; Wang, H.; Wang, W. Optimisation algorithms for spatially constrained forest planning. *Ecol. Model.* **2006**, *194*, 421–428. [CrossRef]
41. Reeves, C. *Modern Heuristic Techniques for Combinatorial Problems*; John Wiley & Sons, Inc.: New York, NY, USA, 1993.
42. Kangas, J. The Analytic Hierachy Process (AHP): Standard Version, Forestry Application and Advances. In *Multiple Use of Forests and Other Natural Resources*; Springer: Dordrecht, The Netherlands, 1999; pp. 96–105. [CrossRef]
43. Shan, Y.; Bettinger, P.; Cieszewski, C.; Wang, W. Pitfalls and potential of particle swarm optimization for contemporary spatial forest planning. *For. Syst.* **2012**, *21*, 468–480. [CrossRef]
44. Ducheyne, E. Multiple Objective Forest Management Using GIS and Genetic Optimisation Techniques. Ph.D. Thesis, Ghent University, Ghent, Belgium, 2003.
45. Pukkala, T. Introduction to Multi-Objective Forest Planning. In *Multi-Objective Forest Planning*; Springer: Dordrecht, The Netherlands, 2002; Section 1, pp. 1–19. [CrossRef]
46. Holland, J.H. *Adaptation in Natural and Artificial Systems*; MIT Press: Cambridge, MA, USA, 1992.
47. Reeves, C. Genetic Algorithms. In *Handbook of Metaheuristics*; Springer: Boston, MA, USA, 2010; Volume 146, pp. 109–139. [CrossRef]
48. Srinivas, M.; Patnaik, L.M. Genetic Algorithms: A Survey. *Computer* **1994**, *27*, 17–26. [CrossRef]
49. Bettinger, P.; Boston, K.; Siry, J.P.; Grebner, D.L. *Forest Management and Planning*; Academic Press: Cambridge, MA, USA, 2016.
50. Jin, X.; Pukkala, T.; Li, F. Fine-tuning heuristic methods for combinatorial optimization in forest planning. *Eur. J. For. Res.* **2016**, *135*. [CrossRef]
51. Bettinger, P.; Graetz, D.; Boston, K.; Sessions, J.; Chung, W. Eight heuristic planning techniques applied to three increasingly difficult wildlife planning problems. *Silva Fenn.* **2002**, *36*, 545. [CrossRef]
52. Pukkala, T.; Kurttila, M. Examining the performance of six heuristic optimisation techniques in different forest planning problems. *Silva Fenn.* **2005**, *39*, 396. [CrossRef]
53. Fonseca, C.M.; Fleming, P.J. Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization. *Icga* **1993**, *93*, 416–423.
54. Horn, J.; Nafpliotis, N.; Goldberg, D.E. Niched Pareto genetic algorithm for multiobjective optimization. In Proceedings of the IEEE Conference on Evolutionary Computation, Orlando, FL, USA, 27–29 June 1994; Volume 1. [CrossRef]
55. Deb, K.; Datta, D.; Fonseca, C.M.; Lobo, F.G.; Condado, P.A.; Seixas, J. Multi-Objective Evolutionary Algorithm for Land-Use Management Problem. *Int. J. Comput. Intell. Res.* **2007**, *3*, 371–384. [CrossRef]
56. Ducheyne, E.; Wulf, R.; Baets, B.D. A Spatial Approach to Forest-Management Optimization: Linking GIS and Multiple Objective Genetic Algorithms. *Int. J. Geogr. Inf. Sci.* **2006**, *20*, 917–928. [CrossRef]
57. Næsset, E. Estimating timber volume of forest stands using airborne laser scanner data. *Remote Sens. Environ.* **1997**, *61*, 246–253. [CrossRef]
58. Kurttila, M. The spatial structure of forests in the optimization calculations of forest planning—A landscape ecological perspective. *For. Ecol. Manag.* **2001**, *142*, 129–142. [CrossRef]
59. Neteler, M.; Mitasova, H. *Open Source GIS: A GRASS GIS Approach*; Springer: New York, NY, USA, 2008; Volume 773. [CrossRef]
60. Matthews, R.M.; Jenkins, T.A.R.; Mackie, E.D.; Dick, E.C. *Forest Yield: A Handbook on Forest Growth and Yield Tables for British Forestry—Forest Research*; Forestry Commission: Bristol, UK, 2016.
61. Diéguez-Aranda, U.; Rojo-Alboreca, A.; Castedo-Dorado, F.; González, J.; Anta, M.; Crecente-Campo, F.; González-González, J.M.; Pérez Cruzado, C.; Rodríguez-Soalleiro, R.; López-Sánchez, C.; et al. *Herramientas Selvícolas Para la Gestión Forestal Sostenible en Galicia*; Xunta de Galicia: Catoira, Spain, 2009.
62. Santos, H.A.; Martins, L. *Ferramentas de Apoio à Gestão do Pinhal-Bravo*; Ministério da Agricultura, Pescas e Alimentação: Lisboa, Portugal, 1986.
63. Patrício, M.S. Análise da Potencialidade Produtiva do Castanheiro em Portugal. Ph.D. Thesis, Universidade Técnica de Lisboa, Instituto Superior de Agronomia, Lisboa, Portugal, 2006.
64. Blank, J.; Deb, K. Pymoo: Multi-Objective Optimization in Python. *IEEE Access* **2020**, *8*, 89497–89509. [CrossRef]

65. Zarei, N.; Azari, A.; Heidari, M.M. Improvement of the performance of NSGA-II and MOPSO algorithms in multi-objective optimization of urban water distribution networks based on modification of decision space. *Appl. Water Sci.* **2022**, *12*, 133. [CrossRef]
66. Verma, S.; Pant, M.; Snasel, V. A Comprehensive Review on NSGA-II for Multi-Objective Combinatorial Optimization Problems. *IEEE Access* **2021**, *9*, 57757–57791. [CrossRef]
67. Konak, A.; Coit, D.W.; Smith, A.E. Multi-objective optimization using genetic algorithms: A tutorial. *Reliab. Eng. Syst. Saf.* **2006**, *91*, 992–1007. [CrossRef]
68. Deb, K.; Sindhya, K.; Okabe, T. Self-adaptive simulated binary crossover for real-parameter optimization. In Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation, London, UK, 7–11 July 2007; Association for Computing Machinery: New York, NY, USA, 2007; pp. 1187–1194. [CrossRef]
69. Hassanat, A.; Almohammadi, K.; Alkafaween, E.; Abunawas, E.; Hammouri, A.; Prasath, V.B.S. Choosing mutation and crossover ratios for genetic algorithms-a review with a new dynamic approach. *Information* **2019**, *10*, 390. [CrossRef]
70. Pham, D.; Karaboga, D. *Intelligent Optimisation Techniques. Genetic Algorithms, Tabu Search, Simulated Annealing and Neural Networks*; Springer: London, UK, 2000. [CrossRef]
71. Jiao, L.; Shang, R.; Liu, F.; Zhang, W. Theoretical basis of natural computation. In *Brain and Nature-Inspired Learning Computation and Recognition*; Elsevier: Amsterdam, The Netherlands, 2020; pp. 81–95. [CrossRef]
72. Nuhu, A.R.; Yan, X.; Opoku, D.; Homaifar, A. A niching framework based on fitness proportionate sharing for multi-objective genetic algorithm (MOGA-FPS). In Proceedings of the 2021 Genetic and Evolutionary Computation Conference Companion, Lille, France, 10–14 July 2021; Association for Computing Machinery, Inc.: New York, NY, USA, 2021; pp. 191–192. [CrossRef]
73. Zitzler, E.; Thiele, L. Multiobjective optimization using evolutionary algorithms—A comparative case study. In *Parallel Problem Solving from Nature—PPSN V, Proceedings of the 5th International Conference, Amsterdam, The Netherlands, 27–30 September 1998*; Eiben, A.E., Bäck, T., Schoenauer, M., Schwefel, H.P., Eds.; Springer: Berlin/Heidelberg, Germany, 1998; pp. 292–301.
74. Zitzler, E.; Brockhoff, D.; Thiele, L. The hypervolume indicator revisited: On the design of Pareto-compliant indicators via weighted integration. In *Evolutionary Multi-Criterion Optimization, Proceedings of the 4th International Conference, EMO 2007, Matsushima, Japan, 5–8 March 2007, Proceedings 4*; Springer: Berlin/Heidelberg, Germany, 2007; pp. 862–876.

Instituto Superior
de Engenharia
Politécnico de Coimbra