SOFTWARE TOOL ARTICLE

REVISED MultiOptForest: An interactive multi-objective optimization tool for forest planning and scenario analysis

[version 2; peer review: 2 approved, 2 approved with reservations, 1 not approved]

Kyle Eyvindson[1,2], Daniel Burgas[3,4], Clara Antón-Fernández[5], Jussi Hakanen[6], Michael Emmerich[6,7], Julian Klein[8], Mikko Mönkkönen[3,4], Tord Snäll[8], Astor Toraño Caicoya[9], Marta Vergarechea[5], Clemens Blattert[10]

[1]Faculty of Environmental Sciences and Natural Resource Management, Norges miljø- og biovitenskapelige universitet, Ås, NO-1432, Norway
[2]Natural Resource Institute Finland (LUKE), Laatokartanonkaari 9, Helsinki, 00790, Finland
[3]Department of Biological and Environmental Science, Jyvaskylan yliopisto, P.O. Box 35, Jyväskylä, 40014, Finland
[4]School of Resource Wisdom, Jyvaskylan yliopisto, P.O. Box 35, Jyväskylä, 40014, Finland
[5]Division of Forest and Forest Resources, Norwegian Institute for Bioeconomy Research, Ås, 1433, Norway
[6]Silo Ai, Fredrikinkatu 57 C, Helsinki, 00180, Finland
[7]Leiden Institute of Advanced Computer Science, Universiteit Leiden, Leiden, South Holland, 2333CA, The Netherlands
[8]Swedish Species Information Centre, Sveriges lantbruksuniversitet, P.O. 7007, Uppsala, 75007, Sweden
[9]Chair of Growth and Yield Science, School of Life Sciences, Technische Universitat Munchen, Hans-Carl-von-Carlowitz-Platz 2, 8354 Freising, Germany
[10]Forest Resources and Management, Swiss Federal Research Institute WSL, Birmensdorf, 8903, Switzerland

v2 **First published:** 21 Jun 2023, **3**:103
https://doi.org/10.12688/openreseurope.15812.1
Latest published: 16 Dec 2024, **3**:103
https://doi.org/10.12688/openreseurope.15812.2

Abstract

MultiOptForest is an open-source software designed to simplify building and solving multi-objective optimization problems for forest planning. It aims to find the optimal portfolio of management regimes that balance the objectives regarding multiple forest ecosystem services and biodiversity. The software flexibly imports data, allowing for the use of a variety of forest simulator outputs. The user provides preference information through a user-friendly graphical interface, where the range of possible values for each objective is provided. MultiOptForest solves the optimization problem producing a set of Pareto optimal solutions, *i.e.,* solutions where none of the objectives can be improved without compromising others. MultiOptForest is versatile enough to design a Pareto optimal forest plan for a small holding to assess management and the trade-off between multiple policy objectives impacting the future development of forests across regions and countries.

Open Peer Review

Approval Status ✔ ✘ ? ? ✔

	1	2	3	4	5
version 2 (revision) 16 Dec 2024				? view	✔ view
version 1 21 Jun 2023	✔ view	✘ view	? view		

1. **Can Vatandaslar**, Artvin Coruh University, Artvin, Turkey
2. **Marc McDill**, Penn State University, State College, USA
   Silvana Nobre, University of Madrid, Madrid, Spain
3. **Olalla Díaz-Yáñez**, Eidgenossische Technische Hochschule Zurich, Zürich,

Keywords
Multi-objective optimization, forest planning, scenario assessment, interactive decision making

H2020 This article is included in the Horizon 2020 gateway.

This article is included in the Forest and Forestry Sciences gateway.

This article is included in the Innovation and Growth in the Bioeconomy collection.

Switzerland
Sina Heubel, Eidgenossische Technische Hochschule Zurich, Zürich, Switzerland

4. **Yuxiang Dong**, Tongji University, Shanghai, China
5. **Logan Robert Bingham**, Technical University of Munich, Hans-Carl-von-Carlowitz-Platz, Germany

Any reports and responses or comments on the article can be found at the end of the article.

Corresponding author: Kyle Eyvindson (kyle.eyvindson@nmbu.no)

Author roles: Eyvindson K: Conceptualization, Formal Analysis, Methodology, Software, Validation, Writing – Original Draft Preparation, Writing – Review & Editing; **Burgas D**: Methodology, Project Administration, Writing – Review & Editing; **Antón-Fernández C**: Data Curation, Methodology, Writing – Review & Editing; **Hakanen J**: Methodology, Visualization, Writing – Review & Editing; **Emmerich M**: Methodology, Software, Writing – Review & Editing; **Klein J**: Data Curation, Methodology, Software, Writing – Review & Editing; **Mönkkönen M**: Funding Acquisition, Methodology, Writing – Review & Editing; **Snäll T**: Funding Acquisition, Methodology, Writing – Review & Editing; **Toraño Caicoya A**: Funding Acquisition, Methodology, Writing – Review & Editing; **Vergarechea M**: Data Curation, Methodology, Software, Writing – Review & Editing; **Blattert C**: Conceptualization, Data Curation, Methodology, Resources, Software, Visualization, Writing – Review & Editing

Competing interests: No competing interests were disclosed.

Grant information: This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 773324 (ForestValue - Innovating forest-based bioeconomy [ForestValue]). This work was also conducted within the project MultiForest, which is supported under the umbrella of 609 ERA-NET Cofund ForestValue by: Academy of Finland, Business Finland, Federal Ministry of Agriculture, Forestry, 610 Environment & Water Management (Austria), Agency for Renewable Resources (Germany), Research Council of 611 Norway, Vinnova (2018-04982; Sweden).
The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

How to cite this article: Eyvindson K, Burgas D, Antón-Fernández C *et al.* **MultiOptForest: An interactive multi-objective optimization tool for forest planning and scenario analysis [version 2; peer review: 2 approved, 2 approved with reservations, 1 not approved]** Open Research Europe 2024, **3**:103 https://doi.org/10.12688/openreseurope.15812.2

First published: 21 Jun 2023, **3**:103 https://doi.org/10.12688/openreseurope.15812.1

REVISED **Amendments from Version 1**

To respond to the reviewers comments we have adjusted the paper as follows:

1. Introduction: We have modified this section to provide additional information surrounding the justification for why our team developed this optimization framework
2. Software description: We have revised the presentation of the software architecture, providing more information regarding the flow of information. We have provided more information on the requirements of the input data, and provide a detailed description of running the software on Code Ocean.
3. Graphical user interface description: We have updated this section to highlight the purpose of the data exportation and visualization of results.
4. Minor typos and general descriptive additions to respond to reviewers comments.

Any further responses from the reviewers can be found at the end of the article

Introduction

Forests play a critical role in providing multiple ecosystem services and biodiversity, which highlights the importance of planning the use of forests at all levels, from small-scale private forest holdings (Pohjanmies *et al.,* 2017) to national or international policy evaluations (Mazziotta *et al.,* 2022). This has led to a rising demand for tools to quantify the conflicts between multiple forest objectives. These tools should be able to investigate the trade-offs and synergies between objectives, in order to plan the silvicultural and harvesting decisions to be taken to meet a wide range of preferences (Linkevičius *et al.,* 2019; Nordström *et al.,* 2019). To meet a wide range of use cases, a general optimization tool needs to be (a) intuitive to use, to allow for a broad range of users with limited computer skills; (b) powerful and able to handle a large variety of problems from forest holdings to national or even international scales; (c) flexible, *i.e.,* can accommodate disparate types of forest growth information provided by different forest simulators; (d) can account for the several categories of objectives that are expected from forests; and (e) transparent, to facilitate evaluation of the delivered results.

With these motivations in mind, we created the MultiOptForest software (Eyvindson *et al.,* 2023), which utilizes a structured optimization process with a strong theoretical foundation. It assists the human decision maker to analyse the huge search space of possible management regimes for preferable Pareto optimal solutions (Borges *et al.*, 2014; Díaz-Yáñez *et al.*, 2021). These are solutions that can no longer be improved in one objective without deteriorating other objectives (Miettinen, 1999). Thereby win-win strategies are always favored whereas all lose-lose strategies are excluded, and in case of trade-off the user can adjust her/his preferences in an interactive dialogue with the software.

The MultiOptForest software builds upon the history of multi-objective optimization frameworks used to create optimized management plans in forestry applications. A few decades ago, forest planning literature often focused on the production of timber resources, aiming for an optimal even-flow of timber over time while minimizing costs of harvesting and logistics (Johnson & Scheurman, 1977). The underlying optimization problems have often been highly structured, focusing on harvest intensity and economic priorities. Nowadays, forests are recognized for their contribution to a wide range of societal demands (Blattert *et al.*, 2022; MEA, 2005; Pascual *et al.*, 2017; Winkel *et al.*, 2022), and this is why there is a need to shift towards a flexible approach accounting for those multi-objectives (e.g. Augustynczik *et al.*, 2020; Eggers *et al.*, 2022; Uhde *et al.*, 2017). The interactive optimization approach used in MultiOptForest is based on optimizing an achievement scalarizing function to find the multi-objective solution closest to the preference information (e.g. demands for ecosystem services) specified by the user. The methodology is what is presented in the desdeo-mcdm package (RRID:SCR_023502) in the DESDEO framework (Misitano *et al.*, 2021), however our software is adapted to forest management planning problems, data and optimization. The MultiOptForest software is also related to the optimLanduse software (Husmann *et al.*, 2022), which optimizes the spatial land use cover compositions to provide a range of ecosystem functions, biodiversity indicators and social preferences. However, optimLanduse uses an robust multi-objective land-cover composition optimization approach of Knoke *et al.* (2016), while MultiOptForest used a multi-objective problem formulation (Miettinen, 1999) in combination with the achievement scalarizing functions approach (Wierzbicki, 1986).

MultiOptForest allows advanced users to connect forest simulator data to an interactive optimization framework that non-expert users (*e.g.*, policy makers, consultants, forest owners, non-governmental organizations) can adjust according to their preferences in an intuitive and transparent manner, without being distracted by the details of the algorithm design and parameters. The MultiOptForest software is designed to use open-source solutions for optimization, although the option to use commercial optimization solvers remains available. The software provides a straightforward approach to construct optimization problem formulations and a systematic method for eliciting preferences from non-expert users.

The input requirements of this software are projections of multiple alternative management trajectories or silvicultural treatments for each forest stand or plot under consideration. This is provided through forest simulation software (Antón-Fernández & Astrup, 2022; Pretzsch *et al.,* 2008; Pretzsch *et al.,* 2002; Rasinmäki *et al.,* 2009; Wikström *et al.,* 2011), where the output represents alternative potential stand-level scenarios dependent on the decisions taken in the forest. The user can define the individual optimization objectives based on the simulator outputs, for instance as an indicator for biodiversity the user may strive to maximize deadwood at a specific year. Additionally, the objective functions should be based on the interest and ability of the forest owner to provide preference information. Determining appropriate optimization objectives currently requires an advanced user, someone who understands both the simulated forest data and the appropriate interpretations of the parameters used to set the optimization objectives. Ongoing development of the software will integrate the construction of objectives with the graphical user interface, allowing more flexibility and usability of the software. Once the objectives are determined and the multi-objective problem has been formulated, the user can interactively provide preference information and explore the corresponding Pareto optimal solutions to better understand the range of Pareto optimal management scenarios and conflicts or trade-off between objectives.

The MultiOptForest software has so far been used to assess and quantify the coherence and incoherence of forest-oriented policies across Fennoscandia (Blattert *et al.,* 2022; Vergarechea *et al.,* 2023) and in Germany (Toraño Caicoya *et al.,* 2023). These studies indicate how to translate forest-related policies into comprehensive optimization problems and show how the identified forest management scenario can lead to meeting the objectives of the policies. As a flexible approach to constructing multi-objective optimization formulations, this software can be used in a variety of forest planning cases, including large scale strategic planning cases (e.g. national forest inventory plot level) as well as small scale tactical and operational planning cases (e.g. individual forest stands of a management unit)(Kangas *et al.,* 2015).

Software description

This optimization framework has been constructed entirely in Python (RRID:SCR_008394), using a variety of open-sourced packages: Pandas, NumPy, MatPlotLib (RRID:SCR_018214, RRD:SCR_008633, RRID:SCR_008624). The backbone package for the optimization is OR-Tools (Perron & Furnon, 2022), which we use to construct the optimization problem. This approach allows for the use of openly available solvers (such as coin-or branch and cut (CBC), coin-or linear programming (CLP), Google linear optimization package (GLOP)) and commercial solvers (such as CPLEX, GUROBI™, XPRESS). For small problems, open solvers are quite capable, but larger problems may require the use of commercial solvers and may require more computational power to conduct the optimizations in a reasonable timeframe. The specific minimum system requirement depends on the size of the input data. However, the example cases found in the source code (Eyvindson *et al.*, 2023, or from Code Ocean) were able to be run on a virtual machine running Ubuntu 18.04, with 3 Gb of ram.

Software architecture

Individual components. The architecture of the software is composed of nine components (Figure 1). The first component focuses on importing simulated forest data. This component converts the data provided to useful format for optimization. The software currently imports data as a large table based on specific structural principles related to forest management options (e.g. effect on ecosystem services and biodiversity). To allow for different forest simulators, this component can be edited to allow for more diverse data types. Data is stored as a table with an index for each forest stand or plot, each forest management option and for each simulated year.

The second component processes the data and constructs the model base for the optimization problem. A framework for the optimization problem is constructed, facilitating the addition of introduction of objectives and constraints. The third component allows for (advanced) users to construct problem specific objectives and constraints for the problem. How this is performed is described further in the sections "Formulating objective functions and epsilon constraints" and "Formulating additional constraints." The fourth component integrates these objectives and constraints into the multi-objective optimization model, following the structure as described in "Multi-objective problem" section. To construct the objectives, the software solves a series of maximization problems for each objective function. This results in a pay-off table (component 5) presenting the ideal and nadir values for each objective, which is used to normalize the objectives and to provide users a reference when providing preference data.

The remaining components interact with preference information provided by the user. Components six and seven relate to the user to quantify their preferences, with component seven being the graphical user interface (GUI) where the user inputs the preference information. When the user inputs preferences they can interact and exploring specific solutions generated by the multi-objective optimization problem through various visualization techniques (Miettinen, 2014). Preference information (component six) is provided as target values (objectives) or minimum threshold values (constraints). Components eight and nine focus on optimizing specific issues based on the preference information from the decision-maker and quantifying the solution for reintroduction into the GUI in component seven. This enables the extraction of solutions that align with the decision-maker's preferences and decision outputs for each forest stand or plot, ensuring a tailored approach to forest management optimization.

Levels of software organization. To enable ease of interaction with the multi-objective optimization component, we constructed a three-layer organization of the code (Figure 2). At the top level, we have the user interface, where the user can provide only the core information required; the forest simulator data and the preferences on ecosystem services. At this level, the user can identify the location of the data and provide preference information for the objectives selected in the multi-objective optimization problem. These objectives are predefined, set by an advanced user in the middle layer of the code.

The second layer of the code allows for more generalizability of the functionalities defined in the third layer of the code. This second layer contains forest simulator-specific information, allowing for flexible interpretation of data, and allows more advanced users to pre-construct objective functions. This

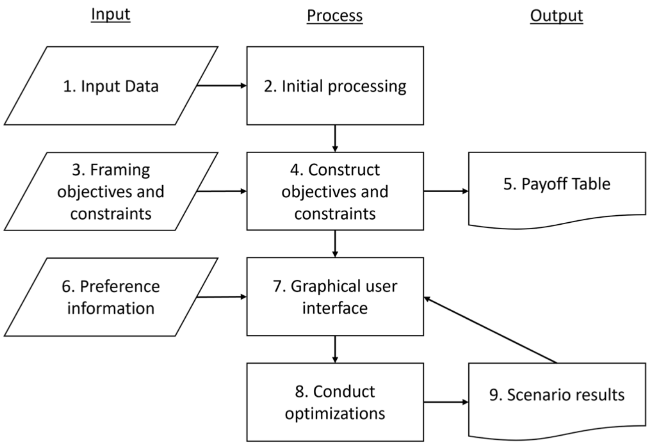

Figure 1. The architecture of the software. The user can input data [component 1], design objectives and constraints [component 3] and provide preference information [component 6]. The model construction, processing of data and output of solutions is automated.

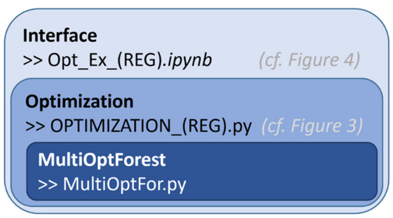

Figure 2. Different levels of the software organized based on complexity of the code. Interface – users can interact with (allows for setting of objectives and constraints). Optimization – code that links the core functions to prepare for the interactive component. MultiOptForest – basic functionality used to conduct the optimization and visualization. REG refers to the regional specific code allowing for minor adaptions to input data and objectives.

will require the user to have a comprehensive understanding of the forest simulation data and the competence to construct python dictionaries according to a predefined template. This code calls specific functions to construct the multi-objective optimization problems in an organized fashion.

The third layer of the code is the core of the software, which is used to construct an object containing the core functions used in optimization and visualization. As the heart of the code, it is designed to flexibly add objective functions, constraints and calculate the key information to construct the multi-objective optimization problem.

Our software has been tested to import data from four forest simulators (SIMO, SiTree, Silva, Heureka), and minor modifications to the second layer of the software will be required to ensure consistency of the data and its interpretation.

Software functionalities

Importing data. Data can be imported from various forest simulators, we have adapted the software to import data from SIMO (Rasinmäki *et al.,* 2009), SiTree (Antón-Fernández & Astrup, 2022), Silva (Pretzsch *et al.,* 2008; Pretzsch *et al.,* 2002), and Heureka (Wikström *et al.,* 2011). MultiOptForest required only minor modifications of the second layer to ensure consistency of the data and its interpretation. These forest simulators provide projections of the development of the forest across time. The projections depend on the forest management regime implemented, which can include the specific timing or intensity of operations (thinning, final felling). The simulations can be at the forest stand level (a relatively homogeneous parcel of forest land) or the forest plot level (a fixed size of forested land, obtained using sampling approaches to represent larger forest areas).

Data can be provided from a variety of different forest simulators, but a minimum standard set of information from all forest simulators is required, indexed data (stand ID, year, management regime) and columns for periodical indicators representing the state of the forest and of the ecosystem services and biodiversity (country or model specific). Those indicators are simulated for each forest entity (forest stand or plot level) under alternative managements by the forest simulators.

Examples for the importing data of the above four forest models are on Code Ocean and can be interactively loaded for the below use case following the step-by-step guideline in the supplementary material.

Formulating objective functions and epsilon constraints. The procedure to create the individual objective functions and epsilon constraints is based on a pre-defined set of options (Listing 1). For any of the forest simulator outputs, it is possible to formulate objectives and epsilon constraints related to temporal development of the specific output. In the current version of the code, there are 12 options for how the temporal aspect is treated (for a list of all, Table 1). For instance, from an ecological perspective, to meet biodiversity targets, we may want to increase the quantity of deadwood in the forest by a specific year, and to ensure that quantity is kept for the rest of the planning horizon. From a timber harvesting perspective, the objective could be to maximize the minimum or to ensure a minimum yearly increase in harvests across the planning horizon. The key differentiation between an objective function, and an epsilon constraint is that a target value is used to identify the preference of an objective, while a threshold value (where anything below is unacceptable) is used for an epsilon constraint). The implementation of each objective is based on eight attributes, linking the data to a specific temporal (c.f. Table 1) and spatial aggregation interpretation, see Listing 1 for examples. Based on these attributes the more advanced user is able to build individual objective functions depending on the optimization targets.

Formulating additional constraints. The majority of the components should be formulated as optimization formulations and epsilon constraints, however there are situations when optimizing an aspect is not needed but only restrictions are required. To facilitate this, our program can generate two types

```
1 OBJ.biodiversity = {
2   #Deadwood - target 2050, increase by XX%
3   "relative_Amount_Deadwood_2050":["Total Deadwood volume by 2050 (%, relative to 2016 values)",
4                                  "Relative_Total_V_total_deadwood”,
5                                  "max","targetYearWithSlope”,”sum”, 2050]}
6 OBJ.wood_production = {
7   #Harvested roundwood - target 2025
8   "Total_Harvested_V_2025":["Total annual harvested timber volume by 2025 (log&pulp) (m3)",
9                                      "Total_Harvested_V",
10                                     "max”,"targetYearWithSlope","sum", 2025]}
```

Listing 1. Two examples defining objective functions relating to forest biodiversity and wood production. The dictionary key is a unique description of the objective (e.g. Total_Harvest_V_2025), and the value of the dictionary is a 6-element list. The elements in the lists starting at lines 3 and 8 are 1) a human-readable string, 2) the column name of the simulation output dataset, 3) “max” or “min” if increase or decrease is aimed for, 4) how to handle temporal aspects, 5) how to handle spatial aspects, 6) the target year or a string of periodic targets (may not be required depending on how the temporal aspects are handled in 4). (cf. Blattert *et al.*, 2022).

Table 1. Advanced users can handle temporal aspects in MultiOptForest when constructing individual objective functions. Each coded interpretation specifies how to implement these temporal aspects when solving the optimization problem.

Function Abbreviation	Plan language interpretation of the temporal aspect
min	Maximize the minimum value over the time horizon
average	Maximize the average value over the time horizon
firstyear	Maximizes the value of the first year
sum	Maximizes the sum of the values over the time horizon
targetYear	Aims to reach a specific target value at a specific year
targetYearWithSlope	Aims to reach a specific target value at a specific year, with a continued linear increase afterwards
lastYear	Maximize the value of the last year
periodicTargets	Meet a specific target value for all years
minYearlyIncrease	Minimize the yearly increase across the time horizon
maxYearlyIncrease	Maximize the yearly increase across the time horizon
minDecreaseDuringNPeriods	Minimize the periodic decrease across the time horizon
maxIncreaseDuringNPeriods	Maximize the periodic increase across the time horizon

of constraints that can 1) restrict management options on specified stands or plots (*e.g.*, protected areas cannot be harvested), 2) require indicators not to exceed a specific reduction (*e.g.*, to avoid species extinction). For example, the first constraint type can be used to limit management on drained peatlands to be only managed using either no management, or continuous cover forestry (Listing 2). The second constraint type can be used to ensure specific threatened species not to go below a specific threshold (see documentation on the git).

Multi-objective problem. The core component of the software is where the optimization problem is constructed. To ease the implementation, we use a theoretically sound multi-objective problem formulation to find efficient solutions for each optimization scenario. This is accomplished using the core of multi-objective optimization (Miettinen, 1999):

$$\underset{x}{\text{minimize}} \{ f_1(x), \ldots, f_n(x) \} \quad \textit{subject } x \in S \tag{1}$$

where Equation 1 describes an optimization that aims to simultaneously minimize a set of n objective functions (or simply ‘objectives’) $f_i(x)$, $i = 1, \ldots, n$, with x being the decision vector for the management regimes, and S is the decision space, *i.e.*, the set all feasible management regimes. In this formulation, all the objectives are to be minimized. If some objective is to be maximized, it is equivalent to minimize $-f_i$.

```
1 OBJ.mfo.defineObjectives(OBJ.objectives,initialValues=OBJ.initialValues)
2 OBJ.CCFregimes = [regime for regime in OBJ.mfo.regimes if "CCF" in regime] + ["SA"]
3 OBJ.constraintTypes = {"CCFonPeat":
4           ["Allowed regimes", "Only CCF on peat lands", OBJ.CCFregimes, "PEAT"]}
5 OBJ.mfo.defineConstraint(OBJ.constraintTypes)
```

Listing 2. Example of an enabled constraint that guarantees that only certain managements (different variants of continuous cover forestry [CCF] and set aside [SA]) are allowed on forest stands situated on peatland ("PEAT"). The dictionary key is a unique description of the constraint, and the value of the dictionary is a 4 or 5-element list, depending on the constraint type. For the "Allowed regimes" constraint type: The elements on line 4 are 1) the constraint type, 2) a human-readable string, 3) the management regimes allowed and 4) the column that identifies the stands or plots where the restriction occurs. For other constraint types, the variables of the list will vary slightly.

The technical implementation of the optimization was accomplished using two components. The first component uses the achievement scalarizing function of Wierzbicki (1986). This component uses reference points to quantify the preferences between the defined objectives. The second component uses the ε-constraint method (Miettinen, 1999), which sets strict requirements for the specific objective and can be interpreted as a maximal (or minimal) level for each objective. Our software will construct both the achievement scalarizing function and epsilon constraint for each objective defined in the 'formulating objective functions' section.

The generalized formulation of the combined multi-objective problem formulation is a combination of the achievement scalarizing function to be minimized (Hartikainen *et al.,* 2016), while incorporating the ε-constraint method:

$$\min_{x} \max_{1\le i\le n} \left(f_i(x) - z_i^{ref}\right) / \left(z_i^{ideal} - z_i^{nadir}\right) + \rho \sum_{i=1}^{n} f_i(x) / \left(z_i^{ideal} - z_i^{nadir}\right) \quad (2)$$

subject to:

$$f_i(x) \le \varepsilon_i,\ i = 1,\dots,n \qquad x \in S \quad (3)$$

Equation 2 evaluates the distance away from ideal vector $z^{ideal} \in R^n$ consists of the optimal values for each objective when optimized individually while the nadir vector $z^{nadir} \in R^n$ consists of the worst values for each objective. The reference point $z^{ref} \in R^n$ is constructed by using the user-defined aspiration levels, the ideal vector and the nadir vector depending on the preferences for the objectives. The second half of the equation is an augmentation term that guarantees that the solutions are Pareto optimal, with ρ set as an arbitrary small positive constant (the choice of the value may depend on the software, solver or other technical choices). Equation 3 is the ε-constraint so that each objective meets or exceeds the value from vector ε. An interesting technical detail is that the achievement scalarization function is not using the Euclidean distance to the reference point, but the Chebychev distance to the reference point. This choice leads to a more balanced approximation of the aspiration levels for the different objectives.

Graphical user interface

The graphical user interface consists of three sections (Figure 2). In the first section, named "Constraint values", users can provide epsilon constraint values (requiring the solution to have values higher than the target). In the second section, named "Reference point", users can provide aspiration levels for each objective (desired value for each objective). The aspiration levels jointly constitute a reference point from which a measure of preferability can be made. In the third section, "Enabled constraints", users can include specified constraints to the decision problem (cf. Figure 3). In the first and second sections ("Constraint values" and "Reference point"), values can be set either by slider bars (see "formulating objective functions") or by typing the desired reference directly in the box (Listing 2). In the third section, constraints can be enabled with a check box (see "formulating constraints"). The ranges of the slider bars on the first and second sections are set to the anti-ideal and ideal values of each indicator, to allow the decision maker to set preferences that are feasible for the specific problem. To engage the additional defined constraints (Enabled constraints) requires a two-step process, first to click the check box and then to click the "Change constraints" box. Once the optimization is completed, options are available to export and visualize the results.

An example how the GUI can be loaded for the below described use case is described in the step-by-step user guideline for Code Ocean in the supplementary material (also linked on the Readme.md instructions).

Data exportation means constructing text files for both the aggregate level and the stand level. The output is primarily a very large table indicating the management option selected for each stand, with a column to indicate how much of the stand is managed according to each specific management. This table includes all indicators produced by the forest simulator under the optimal management for each entity. This allows to produce a wide range further analysis and visualizations (comparable to Figure 4). To aid visualization, we currently have coded a selection of plots for optimal management portfolios and objective achievement plots (comparable Figure 4) as well as simple temporal line graphs, which can be explored on Code Ocean (see Supplementary information for a guideline of how to run the capsule in Code Ocean)

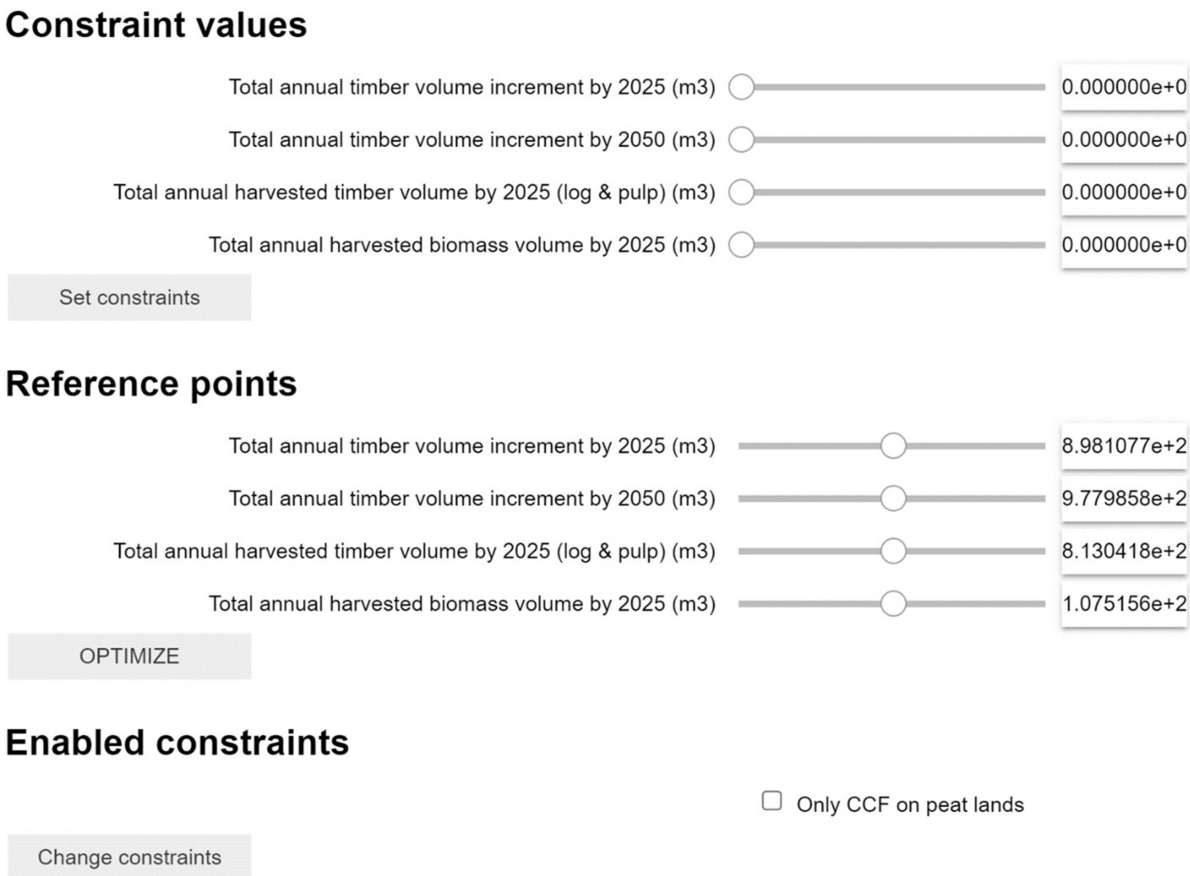

Figure 3. Example of the GUI for an optimization problem aiming to optimize forest management to reach four objectives. Each objective can be addressed as Constraint or Reference Point (or both) depending on the user's needs. Additionally, specific advanced constraints can be activated with the checkbox "Enabled constraint" (see 2.2.3).

Use case

The optimization tool was recently used to develop forest landscape management scenarios that match the multiple societal demands for forest ecosystem services stated in sectoral policies in Finland (Blattert *et al.,* 2022), Norway (Vergarechea *et al.,* 2023) and Germany (Toraño Caicoya *et al.,* 2023). The modular character of the tool allowed to address the diverse objectives of national policies in a flexible way. For illustration, we describe one optimization problem that reflects the societal demands of the Finnish Bioeconomy Policy (FMME *et al.*, 2014).

The Finnish Bioeconomy Policy anticipates a need for increased roundwood and biomass extraction to offset fossil fuels as a means to mitigate CO_2 emissions that leads to a warming climate (FMME *et al.*, 2014). The overall aim of the policy is to simultaneously mobilize forest resources for bioeconomy purposes while safeguarding biodiversity. The forest optimization problem strives to maximize an even-flow of harvested roundwood and biomass under the constraint that biodiversity indicators should not decline (Table 2). Even flow was addressed by maximizing the minimum indicator value over all simulated years. Further, the objectives for the recreational value of forests were maximized as the policy also emphasized the importance of forest recreational value for society. Input data for the optimization was a Finland-wide systematic sample of forest stands (FFC, 2021) that had been simulated with alternative management regimes representing even-aged rotation forest with final clearcut (data are available here) intensified and extensified versions of rotation forestry, regimes that foster adaptation against climate change, continuous cover forestry regimes and under set aside (without any intervention). The simulations were done using the SIMO forest simulation software and projected 100 years into the future under alternative climate scenarios (here presented for illustration purposes only for once climate scenario).

According to the results of the optimization, the policy objectives for the Bioeconomy Policy would require that approximately 2/3 of the forests would be managed by practices that include continuous cover forestry regimes and protected areas (Figure 3). The remaining 1/3 of the forest should instead be intensively managed for wood production.

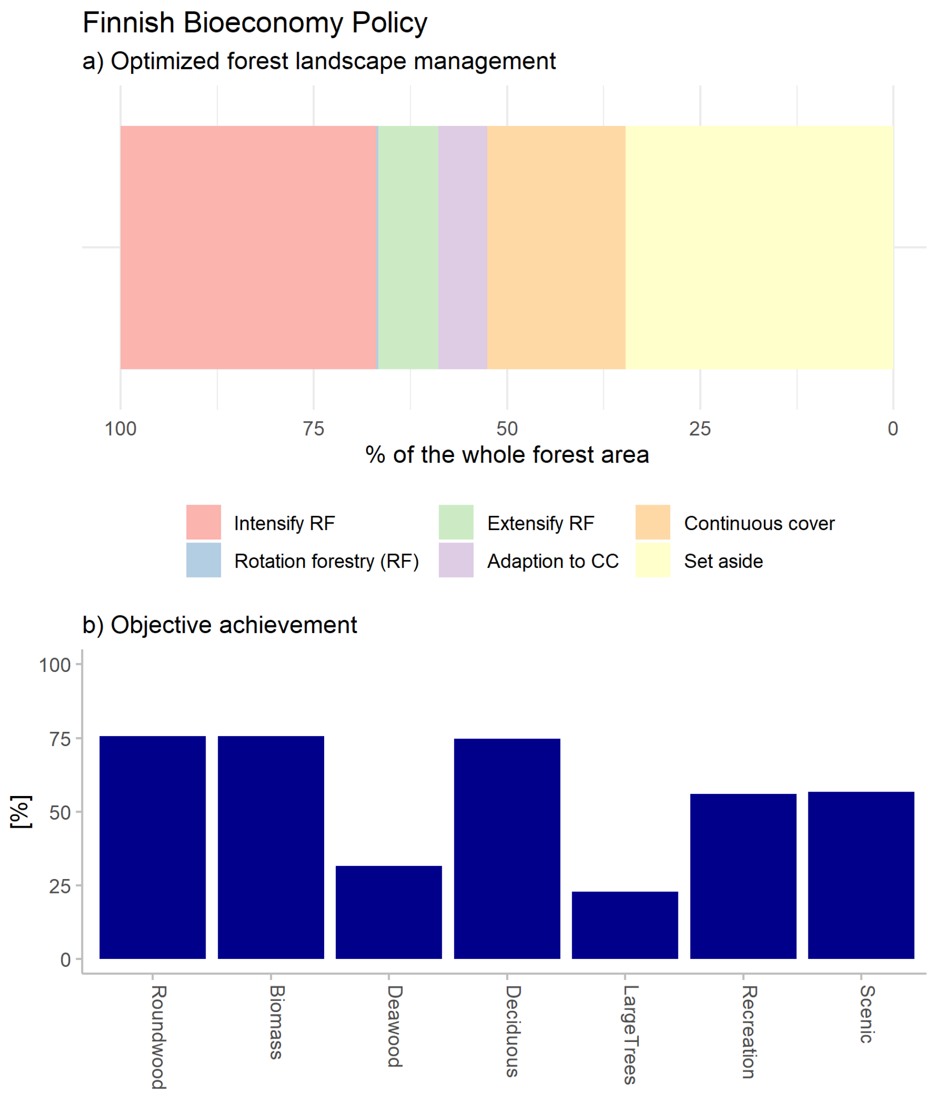

Figure 4. a) Optimal Forest management program that matches the societal demands for forest ecosystem service objectives of the Finnish Bioeconomy Policy under a single climate change scenario; **b**) achievement of the objectives, normalized among the ideal and anti-ideal value (maximum and minimum possible solution) (outcomes of alternative climate and policy scenarios are presented in Blattert *et al.*, 2022).

Table 2. Objectives and constraints that were defined and solved with the MultiOptForest software aiming to elaborate a forest landscape management that would match the societal demands of the Finnish Bioeconomy Policy (cf. (Blattert *et al.*, 2022)).

	Indicator	Objective (Obj) / Constraint (Con)
Wood production	Harvested roundwood (m^3 ha^{-1})	Maximise even-flow **(Obj)**
Bioenergy	Harvested biomass (m^3 ha^{-1})	Maximise even-flow **(Obj)**
Biodiversity Conservation	Deadwood (m^3 ha^{-1}) Deciduous tree volume (%) Large trees (DBH > 40cm) (n ha^{-1})	No decline allowed **(Con)** No decline allowed **(Con)** No decline allowed **(Con)**
Recreation	Recreation index (0-1) Scenic index (0-1)	Maximise **(Obj)** Maximise **(Obj)**

Discussion

The MultiOptForest software simplifies the interactive assessment of complex forest planning challenges. Moreover, the software eases the building of multi-objective optimization problems and integration of new objectives for forest planning. This includes a rich and intuitive interface for formulating objectives and provides a common framework to explore the impact of management and targets for the forest using various sources of input data. The software user interface allows for interactive exploration of the set of Pareto optimal solutions using multi-objective optimization and constraint handling.

The MultiOptForest software was designed in a research project that aimed to critically compare the consistency between forest-related policy documents guiding the national/regional administration and management of forest ecosystem services and biodiversity (Antón-Fernández *et al.,* 2022). At a national level in Finland (Blattert *et al.,* 2022) and Norway (Vergarechea *et al.,* 2023), at a regional level in Germany (Toraño Caicoya *et al.,* 2023), and international level (Blattert *et al.,* 2023) this software has been used to unveil synergies and conflicts between these government policies guiding forest use. With the help of this software, researchers were able to explore how forest management could be applied to balance conflicting objectives and best meet the stated policy goals in each country. The flexibility and functionalities of the software allowed to design a wide variety of objectives that match the diverse interpretations and contextual considerations of sustainable forest management state in these national policy documents.

This software has great potential to be used in a variety of forest planning and forest policy development research, as the tool constructs objective functions in a systematic fashion and allows for the comparison of alternative multi-objective optimization scenarios. While the optimization framework can use simulation data from multiple sources, this software integrates with nationally/regionally specific forest management software to enable nationally/regionally relevant scenarios. Although the current use of the software has focused on very large-scale problems, it can be applied at smaller spatial scales, such as forest holdings, or certain landscapes. The computational time and complexity however depend on the number of individual optimized entities (forest plots or stands). The software has not yet been used in commercial settings, however forest planners could integrate this as an iterative approach to improve forest planning.

Due to the simplicity of its user interface, the MultiOptForest has the potential to become a tool to help untrained users to understand the conflicts among forest ecosystem services and biodiversity. This could be used, for instance, in participatory workshops seeking consensus solutions where forest stakeholders from different sectors examine the consequences of their preferences. The nested construction from simpler to more complex multi-objective optimization problems also allow to use the tool for educational purposes. Nevertheless, in the current version still certain IT skills are required to use the optimization tool. The provided software capsule on Code Ocean together with the short user guide however aims to ease the first steps with the tool avoiding complex pre-installations.

Conclusions

The key benefit of the MultiOptForest software is the flexibility to define the specific objectives of interest to the decision maker. The software utilizes theoretically sound multi-objective optimization techniques, packaged in a user-friendly software package with which decision-makers can interact. The easy-to-understand interface enables users with no training in coding to modify preferences, evaluate different optimized scenarios and gain an understanding of the trade-offs and synergies involved in their decisions.

Data availability

Source data

Input data for the use case presented (and example cases from Germany, Sweden and Norway) is available in the data folder in: https://doi.org/10.5281/zenodo.7885954. And is originally available from https://doi.org/10.5281/zenodo.6631110 (Blattert *et al.,* 2022a) under a CC-BY 4.0 license.

Software availability

Software available from:

Source code: https://codeocean.com/capsule/2689439

Archived source code at the time of publication: https://doi.org/10.5281/zenodo.7885953 (Eyvindson *et al.,* 2023)

License: MIT

Acknowledgements

We acknowledge the helpful discussions with all participants of the MultiForest project, especially Markus Hartikainen for coding work in the software and Johanna Lundström for the data collection and testing of the software.

References

Antón-Fernández C, Astrup R: **SiTree: a framework to implement single-tree simulators.** *SoftwareX.* 2022; **18**: 100925.
Publisher Full Text

Antón-Fernández C, Blattert C, Burgas D, *et al.*: **Better forest policies and management for sustainability transformations: how can expectations for forests be met?** *JYU Reports.* 2022; 1–6.
Publisher Full Text

Augustynczik ALD, Gutsch M, Basile M, *et al.*: **Socially optimal forest**

management and biodiversity conservation in temperate forests under climate change. *Ecol Econ.* 2020; **169**: 106504.
Publisher Full Text

Blattert C, Eyvindson K, Hartikainen M, *et al.*: **Sectoral policies cause incoherence in forest management and ecosystem service provisioning.** *For Policy Econ.* 2022; **136**: 102689.
Publisher Full Text

Blattert C, Mönkkönen M, Burgas D, *et al.*: **MultiOptForest Optimization Notebook (V1.0) (1.0.1).** *Zenodo.* 2022a.
http://www.doi.org/10.5281/zenodo.6631110

Blattert C, Mönkkönen M, Burgas D, *et al.*: **Climate targets in European timber-producing countries conflict with goals on forest ecosystem services and biodiversity.** *Commun Earth Environ.* 2023; **4**: 119.
Publisher Full Text

Borges JG, Garcia-Gonzalo J, Bushenkov V, *et al.*: **Addressing multicriteria forest management with Pareto frontier methods: an application in Portugal.** *For Sci.* 2014; **60**(1): 63–72.
Publisher Full Text

Díaz-Yáñez O, Pukkala T, Packalen P, *et al.*: **Multi-objective forestry increases the production of ecosystem services.** *For Int J For Res.* 2021; **94**(3): 386–94.
Publisher Full Text

Eggers J, Lundström J, Snäll T, *et al.*: **Balancing wood production and biodiversity in intensively managed boreal forest.** *Scand J For Res.* 2022; **37**(3): 213–25.
Publisher Full Text

Eyvindson K, Burgas D, Hartikainen M, *et al.*: **MultiOptForest: an interactive multi-objective optimization tool for forest planning and scenario analysis (Version v1).** *Zenodo.* 2023.
http://www.doi.org/10.5281/zenodo.7885954

FFC: **Finnish Forest Centre, Open forest information.** 2021; [accessed 1 March 2021].
Reference Source

FMME, FMAF, FME: **Finnish Bioeconomy Strategy - Sustainable Growth from Bioeconomy.** Publication of the Finnish Ministry of Employment and Economy (FMEE), Ministry of Agriculture and Forestry (FMAF), Ministry of the Environment (FME), Finland, 30. 2014.
Reference Source

Hartikainen M, Eyvindson K, Miettinen K, *et al.*: **Data-based forest management with uncertainties and multiple objectives.** In: Pardalos, P.M., Conca, P., Giuffrida, G., Nicosia, G. (Eds.), *Machine Learning, Optimization, and Big Data, Lecture Notes in Computer Science.* Springer International Publishing, Cham, 2016; 16–29.
Publisher Full Text

Husmann K, von Groß V, Bödeker K, *et al.*: ***OPTIMLANDUSE*: a package for multiobjective land-cover composition optimization under uncertainty.** *Methods Ecol Evol.* 2022; **13**(12): 2719–2728.
Publisher Full Text

Johnson K, Scheurman H: **Techniques for prescribing optimal timber harvest and investment under different objectives-discussion and synthesis.** *For Sci.* 1977; **23**(suppl_1): a0001–z0001.
Reference Source

Kangas A, Kurttila M, Hujala T, *et al.*: **Decision support for forest management.** *Managing Forest Ecosystems.* Springer International Publishing, Cham, 2015.
Publisher Full Text

Knoke T, Paul C, Hildebrandt P, *et al.*: **Compositional diversity of rehabilitated tropical lands supports multiple ecosystem services and buffers uncertainties.** *Nat Commun.* 2016; **7**(1): 11877.
PubMed Abstract | **Publisher Full Text** | **Free Full Text**

Linkevičius E, Borges JG, Doyle M, *et al.*: **Linking forest policy issues and decision support tools in Europe.** *For Policy Econ.* 2019; **103**: 4–16.
Publisher Full Text

Mazziotta A, Lundström J, Forsell N, *et al.*: **More future synergies and less trade-offs between forest ecosystem services with natural climate solutions instead of bioeconomy solutions.** *Glob Chang Biol.* 2022; **28**(21): 6333–6348.
PubMed Abstract | **Publisher Full Text** | **Free Full Text**

Miettinen K: **Nonlinear multiobjective optimization, International series in operations research & management science.** Kluwer Academic Publishers, Boston, 1999.
Reference Source

Miettinen K: **Survey of methods to visualize alternatives in multiple criteria decision making problems.** *OR Spectr.* 2014; **36**(1): 3–37.
Publisher Full Text

Millennium ecosystem assessment: **Ecosystems and human well-being.** Vol. 5. Island press Washington, DC; 2005.
Reference Source

Misitano G, Saini BS, Afsar B, *et al.*: **DESDEO: The Modular and Open Source Framework for Interactive Multiobjective Optimization.** IEEE Access, 2021; **9**: 148277–148295.
Publisher Full Text

Nordström EM, Nieuwenhuis M, Başkent EZ, *et al.*: **Forest decision support systems for the analysis of ecosystem services provisioning at the landscape scale under global climate and market change scenarios.** *Eur J Forest Res.* 2019; **138**: 561–581.
Publisher Full Text

Pascual U, Balvanera P, Díaz S, *et al.*: **Valuing nature's contributions to people: the IPBES approach.** *Curr Opin Environ Sustain.* 2017; **26–27**: 7–16.
Publisher Full Text

Perron L, Furnon V: **OR-Tools**. 2022.
Reference Source

Pohjanmies T, Eyvindson K, Triviño M, *et al.*: **More is more? Forest management allocation at different spatial scales to mitigate conflicts between ecosystem services.** *Landscape Ecol.* 2017; **32**: 2337–2349.
Publisher Full Text

Pretzsch H, Biber P, Ďurský J: **The single tree-based stand simulator SILVA: construction, application and evaluation.** *For Ecol Manag.* 2002; **162**(1): 3–21.
Publisher Full Text

Pretzsch H, Grote R, Reineking B, *et al.*: **Models for forest ecosystem management: a European perspective.** *Ann Bot.* 2008; **101**(8): 1065–1087.
PubMed Abstract | **Publisher Full Text** | **Free Full Text**

Rasinmäki J, Mäkinen A, Kalliovirta J: **SIMO: an adaptable simulation framework for multiscale forest resource data.** *Comput Electron Agric.* 2009; **66**(1): 76–84.
Publisher Full Text

Toraño Caicoya A, Poschenrieder W, Blattert C, *et al.*: **Sectoral policies as drivers of forest management and ecosystems services: a case study in Bavaria, Germany.** *Land Use Policy.* 2023; **130**: 106673.
Publisher Full Text

Uhde B, Heinrichs S, Stiehl CR, *et al.*: **Bringing ecosystem services into forest planning - Can we optimize the composition of Chilean forests based on expert knowledge?** *For Ecol Manag.* 2017; **404**: 126–40.
Publisher Full Text

Vergarechea M, Astrup R, Fischer C, *et al.*: **Future wood demands and ecosystem services trade-offs: a policy analysis in Norway.** *For Policy Econ.* 2023; **147**: 102899.
Publisher Full Text

Wierzbicki AP: **On the completeness and constructiveness of parametric characterizations to vector optimization problems.** *OR Spektrum.* 1986; **8**: 73–87.
Publisher Full Text

Wikström P, Edenius L, Elfving B, *et al.*: **The heureka forestry decision support system: an overview.** *Math Comput For Nat-Res Sci.* 2011; **3**(2): 87–95.
Reference Source

Winkel G, Lovrić M, Muys B, *et al.*: **Governing Europe's forests for multiple ecosystem services: opportunities, challenges, and policy options.** *For Policy Econ.* 2022; **145**: 102849.
Publisher Full Text

Open Peer Review

Current Peer Review Status: ✔ ✘ ? ? ✔

Version 2

Reviewer Report 10 January 2025

https://doi.org/10.21956/openreseurope.20653.r48766

✔

Logan Robert Bingham

Technical University of Munich, Hans-Carl-von-Carlowitz-Platz, Freising, Germany

The manuscript "*MultiOptForest: An interactive multi-objective optimization tool for forest planning and scenario analysis*" introduces an open-source decision support tool that uses multi-objective optimization to support the incorporation of ecosystem services into forest management planning. The manuscript has already gone through one round of review. The points raised by the other referees struck me as overall useful and relevant; I'll leave it to them to determine whether the revision has resolved their concerns. Instead, I'd like to offer another perspective: I found this manuscript useful for the work I'm doing right now, and I expect the same will be true for others in my field, so I think it should be indexed.

While there's always room to refine the description and add more detail, the main contribution is to introduce software architecture and usage to a primarily scientific readership—not necessarily to undertake an exhaustive theoretical analysis or present a detailed user manual in the main text (the online supplementary materials contain additional detail and example problems). I think it succeeds. Others might reasonably disagree, but given the state of the software and the manuscript, I think it's appropriate for that discussion to move beyond the peer review process and into the broader scientific community. Relatively approachable open-source Pareto frontier tools for forest management are not exactly in great abundance; I think this one should be disseminated so that a wider audience can experiment with different applications, especially since work to further develop the software and expand the GUI is ongoing.

Below, I make a few minor comments the authors can incorporate or not at their own discretion.

Minor comments

1. Manage reader expectations

Several of the reviewers' previous comments involved the usability of the software and how much expertise is required. In the revision, the authors helpfully distinguish between which actions are available to non-expert users and which require some coding knowledge. That's fair, but I have to confess that I also felt a bit confused by this point on my first read. Maybe the first paragraph of the introduction could do a better job of managing reader expectations? The reader gets the impression that the software meets all of the points (a-e) simultaneously, which isn't quite the case. For example, I don't think it's the case that users with limited computer skills (point a) can integrate different growth simulators (point c) or different objectives (point e); these require more advanced scripting and/or forest management knowledge. The authors might consider rephrasing or adding some qualification here to better calibrate reader expectations. I think this might help clear up some of the previous comments.

2. Clarifying the role of expertise

In a similar vein, later in the introduction the authors note that advanced users can connect forest simulator data, while "non-expert" users like policymakers and NGOs can explore scenarios by playing around with the GUI. Having clicked through the example case studies in the Jupyter notebook, I'm not quite sure that "non-expert" is the best designation. While it's true the GUI allows users who can't code to adjust preferences and run the model, it seems to me that some expertise is needed to understand the sliders and the trade-offs in the first place. For instance, the ranges of some sliders are fairly large, and quite small for others, units vary, and interpreting the trade-offs does seem to require some subject matter expertise (even if the user doesn't need to be familiar with Python to manipulate the actual sliders). Maybe the expert vs non-expert distinction would be better phrased as "scripting users vs GUI users" or "technical users vs basic users"? Alternatively, the authors might set aside the issue of user abilities entirely and just focusing on the software itself: i.e., remove the stuff about experts and non-experts and just keep descriptions like "In the current version of the software, the GUI interface allows users to do ABC, but users who are familiar with Python syntax can additionally perform actions XYZ."

3. A few minor typos/language issues

p4 para 1: "... optimal management scenarios and conflicts or trade-off between" should be "trade-offs"

p4 para. 3: RAM usually capitalized

p4 middle right: --> "Components six and seven relate to quantifying user preferences" or something along those lines?

p4 right, Formulating additional constraints: "The majority of the components should be formulated as optimization formulations and epsilon constraints" Do you mean "formulated as objective functions and epsilon constraints"?

p. 8 "Use case": Maybe "The modular character of the tool allows diverse policy objectives to be addressed in a flexible way"?

Fig. 4a caption: "ideal and anti-ideal" Wasn't the term used previously ideal and nadir?\

Is the rationale for developing the new software tool clearly explained?

Yes

Is the description of the software tool technically sound?
Yes

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?
Partly

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?
Yes

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: Ecosystem services, forest management, trade-off analysis, decision support systems

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard.

Reviewer Report 26 December 2024

https://doi.org/10.21956/openreseurope.20653.r48764

?

Yuxiang Dong
Tongji University, Shanghai, Shanghai, China

1. summary of the article

The software proposed demonstrates academic value and overall performs well in its intended purpose. For me the primary contribution of this paper is its attempt to develop an interactive model that engages a broader audience, including non-experts, to explore varying preferences for optimal solutions. Also, the compatibility with multiple simulators and potential optimization solvers represents another innovative aspect.

While the model shows promise, further simplification and enhanced flexibility may be necessary to facilitate its direct application in a broader range of contexts.

I have reviewed not only the current version of the paper but also the reviewer comments from the previous round revision. The paper shows notable improvements, but I'd like to offer some further suggestions for authors' consideration.

2. Detailed suggestions for the question: Are sufficient details of the code, methods and analysis (if

applicable) provided to allow replication of the software development and its use by others?

a. I found several reviewers from the previous round raised concerns about the input requirements, to which the authors responded by adding descriptions. However, I still found them insufficient to fully address the concern. I understand that clarifying the input requirements is challenging at this stage as many variables appear to be hardcoded within the existing codebase. For example, upon reviewing the source code, I found several lines used for defining regime names as follows:

```
# snippets from Optimization_FIN.py
self.regimeClassNames =
{"regimeClass0name":"CCF","regimeClass1name":"SA","regimeClass2name":"Broadleave"}

self.regimeClassregimes =
{"regimeClass0regimes":["CCF_3","CCF_4","BAUwGTR"],"regimeClass1regimes":["SA"],"regimeClass2regimes":["
"BAUwT_5_B", "BAUwT_15_B", "BAUwT_30_B", "BAUwT_GTR_B"]}
# snippets from Optimization_GER.py
self.regimeClassNames = {"regimeClass0name":"CCF","regimeClass1name":"SA"}

self.regimeClassregimes = {"regimeClass0regimes":["CCF_P3
","CCF_P3_p1","CCF_P3_p2"],"regimeClass1regimes":["NOT"]}
```

Similarly, both constraints and objectives are currently hardcoded. Hardcoding may not be a practical or flexible way for a universal tool intended for non-experts. A potential way could involve modifying the code to identify variables automatically from the input files. For example, users could be instructed to reorganize their input table into columns labeled `id(of stands)`, `year`, `regime`, and other targets. Regimes could be formatted to follow specific patterns such as those beginning with `CCF` or `BAU`. Using such patterns, column names can be easily identified as regime or target names in Python, enabling the GUI to automatically update based on the input. Implementing this feature could enhance the model's usability, as many non-experts are likely to prefer uploading their results directly and interacting with the visual interface without additional coding tasks.

I understand this tool is still under development, and may not be updated immediately, so I suggest the authors either discuss universal input processing as a future task or provide justification for retaining the tool in its current form.

b. Traditionally, a posteriori methods require obtaining the entire Pareto frontier, whereas interactive methods seem to adopt a logic that lies between a priori and a posteriori approaches. Some readers, myself included, may initially wonder why the authors have not provided a tool for plotting the entire Pareto frontier. I suggest that the authors briefly introduce the theory behind the interactive multi-objective optimization method and explain how it differs from traditional a priori and a posteriori methods in the introduction. Additionally, it may be helpful to provide a clear explanation in the main text on how Pareto solutions and trade-offs can be explored using the interactive method, as this objective was mentioned in the introduction.

3. Minor issues

a. The section named "formulating objective functions and epsilon constraints", is sometimes referred as section "formulating objective functions". Same for section named "formulating additional constraints". Please make them consistent.

b. The abbreviation 'DBH' in table 2 was not mentioned and explained earlier in the text.

c. The conclusion describes the tool as "packaged in a user-friendly software package," which may be an overstatement. A Jupyter Notebook-based GUI might not fully qualify as a packaged software. I suggest revising this statement for greater accuracy and rigor.

Is the rationale for developing the new software tool clearly explained?
Yes

Is the description of the software tool technically sound?
Yes

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?
Partly

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?
Yes

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: Multi-objective optimization, ecosystem service, green infrastructure

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however I have significant reservations, as outlined above.

Version 1

Reviewer Report 02 November 2023

https://doi.org/10.21956/openreseurope.17082.r35235

? **Olalla Díaz-Yáñez**

Forest Resources Management - Department of Environmental Systems Science, Eidgenossische Technische Hochschule Zurich, Zürich, Zurich, Switzerland
Sina Heubel
Forest Resources Management - Department of Environmental Systems Science, Eidgenossische Technische Hochschule Zurich, Zürich, Zurich, Switzerland

The software tool article "MultiOptForest: An interactive multi-objective optimization tool for forest planning and scenario analysis" describes the open-source software MultiOptForest. This software is designed to simplify the process of building and solving multi-objective optimization problems that consider various ecosystem services. The tool has been tested in several case studies in Fennoscandia and Germany.

Even though there are several software applications available that can provide Pareto optimal solutions and illustrate trade-offs among multiple objectives, most of these software tools are either not open-source or lack clear instructions regarding their usability. Therefore, we strongly believe that there is a clear justification for creating a new software tool that is open-source and can be easily accessed by all.

We think that the software tool article and its related materials hold academic value, although we have a few minor suggestions that we would like the authors to consider. Specifically, we have some comments and suggestions on how the software's description could be improved. Additionally, we believe that some aspects of the article need further elaboration, as they are not clear enough from the current text. Please refer to our detailed comments below for further clarification.

The article provides detailed information about the code and methods used in the software, which can be replicated by others. It also includes four examples that were used in other publications as supplementary materials. Although the authors have provided a text readme, we believe that a visual step-by-step guide, such as a vignette, could be added to facilitate the understanding of these examples.

We also think that a more detailed description of the software code in the article text, such as adding a pseudocode, would benefit the readers. This will help them understand the different parts of the routine, the flow between the different steps, and explain some aspects of the text more clearly.

The information provided allows interpretation of the approach and results via one example in the article and more in the appendix materials. However, we would appreciate it if the example in the text could be explained more thoroughly. Please refer to our detailed comments below for further clarification.

Detailed comments:

Rationale: The authors of the paper mentioned the importance of having a general optimization tool that is easy to use, even for users who have limited computer skills. However, they also acknowledged that to use this tool, advanced user skills are required, such as for determining appropriate optimization objectives. Although a user interface can help visualize trade-offs under different preferences, there are still previous steps in using this tool that are not intuitive for all users. We recommend that the discussion section address this point as well.

Description of the software tool - The input requirements: In the description of the software tool, it was initially challenging to understand the input requirements needed by the optimization tool when they were first introduced in the paragraph: “The input requirements of this software are projections of multiple alternative management...”. To make it clearer, a concrete example could be added early on, perhaps in the same paragraph, with a reference to the section where the formulation of objective functions is described. For instance, "The user can define the individual optimization objectives based on the simulator outputs, e.g., for biodiversity, the user may choose the relative amount of deadwood at a specific year."

Description of the software tool - The eight standardized attributes: We want to make sure we understand correctly: the user is required to calculate standardized attributes based on the outputs of their simulators. The reason for the minimum standard set of information from the forest simulators is to enable the estimation of the eight selected attributes, correct? We believe that further clarifying these two aspects would benefit the reader.

Description of the software tool - The preference information: Is the preference information provided as a weight to the objective function? Is the user defining these weights? Please clarify this also in the text.

Example provided: In the article, an example is presented, and the results are displayed in Figure 3, which illustrates the Finnish case study. Although this example is from a published study, it would be helpful to provide more information about the context. A more detailed caption could be added, specifying whether the optimization was spatially explicit or conducted under different climate scenarios. It is unclear how the user of the tool can analyze the Pareto optimal solution space from this example. It would be beneficial to see this aspect of the tool's functionality demonstrated in the example, as it is one of the tool's strengths.

Minor comments:
Text:

- “however for the example cases found in the source code (Eyvindson et al., 2023, or from Code Ocean) were able to be run with a virtual machine running Ubuntu 18.04, with 3 Gb of ram.” → Grammar → "However, the example cases found in the source code (Eyvindson et al. (2023), or from Code Ocean) were able to be run on a virtual machine running Ubuntu 18.04 with 3 GB of RAM."

- “Our software has been tested to import data from four forest simulators” → add which forest simulators

- “Data can be provided from a variety of different forest simulators, but a minimum standard set of information from all forest simulators is required, indexed data (stand ID, year, management regime) and indicators representing the state of the forest and ecosystem services (country or model specific)” → The period at the end of the sentence is missing

Figures, Listings, and Tables:

- **Figure 1:**
  - What does this part of the caption refer to and what does it mean in the caption of this

figure: “T2.1 Accounting for risks and uncertainties in forest-based businesses, sectoral projections, and policy design.”

- In box “interface”: reference to Figure 4 but Fig 4 not available in article → should be Figure 2 (?)

- In box “Optimization”: reference to Figure 2 → example for GUI rather than optimization code

- **Table 1:** Improve this caption: “Table 1. Approaches in how the temporal aspects can be handled in the MultiForestOpt software, for each coded interpretation of how temporal aspects are currently implemented.”

Other: The data and descriptions thereof for the applied Finnish use case are not available in English, which makes it more complicated to reproduce and understand.

Is the rationale for developing the new software tool clearly explained?

Yes

Is the description of the software tool technically sound?

Partly

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?

Yes

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?

Partly

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: Forest dynamics, forest optimization, forest management and planning

We confirm that we have read this submission and believe that we have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however we have significant reservations, as outlined above.

Author Response 02 Dec 2024

Kyle Eyvindson

Authors response:

We appreciate the constructive review. To respond to the suggestions and comments, we respond to the reviewer directly. **Reviewer comments:**

The software tool article “MultiOptForest: An interactive multi-objective optimization tool for forest planning and scenario analysis” describes the open-source software MultiOptForest. This software is designed to simplify the process of building and solving multi-objective optimization problems that consider various ecosystem services. The tool has been tested in several case studies in Fennoscandia and Germany. Even though there are several software applications available that can provide Pareto optimal solutions and illustrate trade-offs among multiple objectives, most of these software tools are either not open-source or lack clear instructions regarding their usability. Therefore, we strongly believe that there is a clear justification for creating a new software tool that is open-source and can be easily accessed by all. We think that the software tool article and its related materials hold academic value, although we have a few minor suggestions that we would like the authors to consider. Specifically, we have some comments and suggestions on how the software's description could be improved. Additionally, we believe that some aspects of the article need further elaboration, as they are not clear enough from the current text. Please refer to our detailed comments below for further clarification.

Authors response:

Thank you for the constructive comments, we generally agree with the vast majority of the suggestions, and have integrated them when possible into the article. **Reviewer comments:**

The article provides detailed information about the code and methods used in the software, which can be replicated by others. It also includes four examples that were used in other publications as supplementary materials. Although the authors have provided a text readme, we believe that a visual step-by-step guide, such as a vignette, could be added to facilitate the understanding of these examples. **Authors response:**

When responding to Reviewer 2, we have updated the software architecture section of the manuscript to better describe the flow and steps of the optimization process. We hope that this adjustment also satisfies this reviewer. Additionally, we have generated a step-by-step user guide illustrating with screen shots how the tool can be virtually launched via Code Ocean, providing as such an illustrative example how user can get familiar with the tool: How to start the tool on Code Ocean, how to Launch Jupiter notebook, selecting one of the country example, and run the Jupyter Notebook, which loads the GUI where constraints/objective can be set. (see also our response to Reviewer #1). The step-by-step user guide can be found in the supplementary material and is linked to the Readme.md on Code Ocean.

Reviewer comments:

We also think that a more detailed description of the software code in the article text, such as adding a pseudocode, would benefit the readers. This will help them understand the different parts of the routine, the flow between the different steps, and explain some aspects of the text more clearly. **Authors response:**

We have revised the software architechture section including a new section on “Individual components” followed by the exisitng part under the new defined section “Levels of software organization” (see also our response to R#2). A more detailed description of the software code we have refrained, as we believe this would make the article to complex. All code is transparent and open source via code ocean allowing even an “online exploration of the funktionalities without any software installtions, as well as on the git Repository.

For the flow between the different steps we created a detailed step-by-step user guide for Code Ocean, which can be found in the Supplementary material (see our comment above).

Reviewer comments:

The information provided allows interpretation of the approach and results via one example in the article and more in the appendix materials. However, we would appreciate it if the example in the text could be explained more thoroughly. Please refer to our detailed comments below for further clarification.

Authors response:

To improve the description of the software code, we have thoroughly updated Figure 1 – which now presents the architecture of the software. While this isn't exactly Pseudocode – it does more clearly highlight what is happening with the different components of the code.

Reviewer comments:

Detailed comments:

Rationale: The authors of the paper mentioned the importance of having a general optimization tool that is easy to use, even for users who have limited computer skills. However, they also acknowledged that to use this tool, advanced user skills are required, such as for determining appropriate optimization objectives. Although a user interface can help visualize trade-offs under different preferences, there are still previous steps in using this tool that are not intuitive for all users. We recommend that the discussion section address this point as well.

Authors response:

We agree that there is a combination of advanced user skills required – which is required to adjust the code to enable data imputation and modifications to the selected objective functions. However, from a user experience, we aim to ease the optimization process as much as possible. We reflect on this in the discussion section. See the last sentence of the last paragraph in the discussion.

Reviewer comments:

Description of the software tool - The input requirements: In the description of the software tool, it was initially challenging to understand the input requirements needed by the optimization tool when they were first introduced in the paragraph: "The input requirements of this software are projections of multiple alternative management…". To make it clearer, a concrete example could be added early on, perhaps in the same paragraph, with a reference to the section where the formulation of objective functions is described. For instance, "The user can define the individual optimization objectives based on the simulator outputs, e.g., for biodiversity, the user may choose the relative amount of deadwood at a specific year."

Authors response:

We agree with this, and we hope that the restructuring of figure 1 helps with this. Additionally, we have revised the input requirements in the introduction section to: "The user can define the individual optimization objectives based on the simulator outputs, for instance as an indicator for biodiversity the user may strive to maximize deadwood at a specific year. Additionally, the objective functions should be based on the interest and ability of the forest owner to provide preference information." Additionally, we aimed to highlight the advantages of the open source example provided on Code Ocean, which aims to ease first application steps (see also comments above)

Reviewer comments:

Description of the software tool - The eight standardized attributes: We want to make sure we understand correctly: the user is required to calculate standardized attributes based on the outputs of their simulators. The reason for the minimum standard set of information from the forest simulators is to enable the estimation of the eight selected attributes, correct? We believe that further clarifying these two aspects would benefit the reader.

Authors response:

The attributes are more related to the 'advanced user', although we are working on constructing a GUI component where a normal user can construct the objectives. We currently limit the construction of objectives to more advanced users to ensure that the logic of the objectives are reasonable. The set of information is mainly to define how the objective is constructed. The user doesn't need to standardize anything – this is done through the payoff tables that were constructed in the code. We removed the word "standardized" to avoid misunderstandings. Additionally, we added the sentence, that "Based on these attributes the more advanced user is able to build individual objective functions depending on the optimization targets."

Reviewer comments:

Description of the software tool - The preference information: Is the preference information provided as a weight to the objective function? Is the user defining these weights? Please clarify this also in the text.

Authors response:

The preference information is provided as reference values and shouldn't be interpreted as weights. We have opted to use the achievement scalarizing function, that uses reference points as a direction. The reference points should be interpreted as "the value of the specific objective where the user is satisfied". We have opted to not use weights, as it can be difficult to understand what the weights mean in an optimization sense.

When updating Figure 1, we have discussed what the preference information is "provided as target values (objectives) or minimum threshold values (constraints)"

Reviewer comments:

Example provided: In the article, an example is presented, and the results are displayed in Figure 3, which illustrates the Finnish case study. Although this example is from a published study, it would be helpful to provide more information about the context. A more detailed caption could be added, specifying whether the optimization was spatially explicit or conducted under different climate scenarios. It is unclear how the user of the tool can analyze the Pareto optimal solution space from this example. It would be beneficial to see this aspect of the tool's functionality demonstrated in the example, as it is one of the tool's strengths.

Authors response:

We have added some additional information into the caption of figure 4 (now after revisions), and in the text. For this case, we are only exploring a single future scenario (so one climate change scenario), without uncertainty.

Figure 4 does show the results of one case study and one climate scenario, and doesn't assess trade-offs/Pareto frontiers between objectives. This wasn't the focus of the optimization approach, nor was this the focus of the software. However, this software could be used to assess trade-offs, if desired. This would however require storing a variety of solutions to enable comparisons. To visualize trade-offs between preference sets, you would need to extract solutions for each preference set, and compare. However, from our perspective, multi-objective optimization was used to construct preference specific solutions, and not specifically to quantify and evaluate trade-offs. (please see also our response to R#2 comment 4)

Reviewer comments:

Minor comments:

Text:

- "however for the example cases found in the source code (Eyvindson et al., 2023, or from

Code Ocean) were able to be run with a virtual machine running Ubuntu 18.04, with 3 Gb of ram.” → Grammar → "However, the example cases found in the source code (Eyvindson et al. (2023), or from Code Ocean) were able to be run on a virtual machine running Ubuntu 18.04 with 3 GB of RAM."
• “Our software has been tested to import data from four forest simulators” → add which forest simulators
• “Data can be provided from a variety of different forest simulators, but a minimum standard set of information from all forest simulators is required, indexed data (stand ID, year, management regime) and indicators representing the state of the forest and ecosystem services (country or model specific)” → The period at the end of the sentence is missing
Figures, Listings, and Tables:
• Figure 1:
- What does this part of the caption refer to and what does it mean in the caption of this figure: “T2.1 Accounting for risks and uncertainties in forest-based businesses, sectoral projections, and policy design.”
• - In box “interface”: reference to Figure 4 but Fig 4 not available in article → should be Figure 2 (?)
• - In box “Optimization”: reference to Figure 2 → example for GUI rather than optimization code
• Table 1: Improve this caption: “Table 1. Approaches in how the temporal aspects can be handled in the MultiOptForest software, for each coded interpretation of how temporal aspects are currently implemented.” **Authors response:** We agree, and have edited these minor adjustments as required. We have included a description of the forest simulators used in the development of this software. We have adjusted the figures to refer to the correct aspects. And have improved the caption of table 1 to: ”Table 1. Advanced users can handle temporal aspects in MultiOptForest when constructing individual objective functions. Each coded interpretation specifies how to implement these temporal aspects when solving the optimization problem” **Reviewer comments:**
Other: The data and descriptions thereof for the applied Finnish use case are not available in English, which makes it more complicated to reproduce and understand. **Authors response:**
Example and data on code ocean are in English.

Competing Interests: No competing interests were disclosed.

Reviewer Report 02 November 2023

https://doi.org/10.21956/openreseurope.17082.r35230

✗ **Marc McDill**

Penn State University, State College, PA, USA

Silvana Nobre

ETS Ingeniería de Montes, Forestal y del Medio Natural, University of Madrid, Madrid, Spain

The article describes a software system that can be used to formulate and solve multi-objective forest management problems. The software is undoubtedly an excellent product for the developers to use for their research, but the article does not provide sufficient information for new users who might be interested in using the software. Furthermore, the article would not be particularly helpful to developers interested in creating similar applications.

Major comments: System "Architecture"

The article identifies two types of users of MultiOptForest: advanced users and non-expert users. It seems that most applications would involve both types of users. It would be helpful to have some kind of flow chart that shows the process of building a MultiOptForest problem instance and what the roles are of these different users. Such a flow chart should start from the beginning (creating input files) and guide the reader through to the point where outputs are produced. It would indicate the inputs and outputs from the system, where and how users provide input, where and how output is provided, and what happens in between. This is what I think "software architecture" is about. Figure 1, unfortunately, does not help with understanding these things.

Regarding inputs, it would be helpful to describe more specifically what inputs are required and how they are input into the system. The article is very vague on this point, indicating only that one would need and "advanced user" to understand these things. The article should indicate what inputs are provided through files that are read by the system and how these files must be structured. Also, which inputs are provided through the GUI, and at what point in the process does this occur? There is a paragraph on importing data, but this paragraph does not indicate specifically what data are needed by the system, only what kind of systems have been used to generate input data. If I wanted to use a different system (or even one of those listed) what would I need to do to create input files for MultiOptForest?

In the middle of the system (somewhere) an (or more likely many...) optimization problem is formulated and solved. The article discusses how different solvers can be used, but, again, what would I have to do if I wanted to use Gurobi - for instance - within MultiOptForest?

On the output side, the article offers three vague sentences about the program's output. I would like to know more about what is included in the output. Does it look something like Figure 3? How might the output look for a different type of forest, perhaps with different management options and different objectives? What if I want more detail about a given Pareto optimal solution? How can I visualize trade-offs among different preference sets? Figure 3 shows results for one set of preferences from the case study, but it does not illustrate how the system was used to assess trade-offs between different objectives, which, to me, is a key point of multiobjective optimization.

Major comments: Constraints vs Objectives

The article is not clear on the distinction between constraints and objectives. For example, one item listed as an objective in the section "Formulating objective functions" sounds more like a constraint than an objectives. In particular, "increase the quantity of deadwood in the forest by a specific year, and to ensure that quantity is kept for the rest of the planning horizon" sounds like a constraint to me. Furthermore, the software offers a dozen ways to create an objective but only (apparently) two ways to create a constraint. In the case study (see Table 2), objectives and constraints are mixed together with no indication which items are objectives and which are constraints.

Minor comments:

The article states that: "Forest planning literature often focuses on the production of timber resources, obtaining a relative even-flow of timber over time while minimizing costs of logistics and maximizing timber production." This is not accurate. There is a large literature of forest planning models that address multiple objectives.

The article references two similar packages, desdeo-mcdm and optimLanduse, but it does not elaborate on what the similarities and differences are between MultiOptForest and these other packages. For example, could one use either desdeo-mcdm and optimLanduse instead of MultiOptForest?

I would like to acknowledge the assistance of Silvana Nobre in developing this review.

Is the rationale for developing the new software tool clearly explained?
Yes

Is the description of the software tool technically sound?
Partly

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?
No

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?
Partly

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: Forest economics, forest management, forest management planning, optimization, forest ecology, remote sensing of forests

We confirm that we have read this submission and believe that we have an appropriate level of expertise to state that we do not consider it to be of an acceptable scientific standard, for reasons outlined above.

Author Response 02 Dec 2024

Kyle Eyvindson

Authors response: We appreciate the constructive review. To respond to the suggestions and comments, we respond to the reviewer directly.

Reviewer comments: The article describes a software system that can be used to formulate and solve multi-objective forest management problems. The software is undoubtedly an excellent product for the developers to use for their research, but the article does not provide sufficient information for new users who might be interested in using the software. Furthermore, the article would not be particularly helpful to developers interested in creating similar applications. Major comments: System "Architecture"
The article identifies two types of users of MultiOptForest: advanced users and non-expert users. It seems that most applications would involve both types of users. It would be helpful to have some kind of flow chart that shows the process of building a MultiOptForest problem instance and what the roles are of these different users. Such a flow chart should start from the beginning (creating input files) and guide the reader through to the point where outputs are produced. It would indicate the inputs and outputs from the system, where and how users provide input, where and how output is provided, and what happens in between. This is what I think "software architecture" is about. Figure 1, unfortunately, does not help with understanding these things.
Authors response: We thank the reviewer for this constructive feedback. When constructing the 'architecture', we were considering the levels of abstraction that were being made in the code – however, as the reviewer indicates is not very informative. To address this, we have revised figure 1, to reflect the architecture of the software in a more sensible fashion, highlighting the core components of the optimization tool. Those we grouped along: the required input, processing the data and output from the optimization framework, with the key tasks performed at each step. We have also revised software architecture section to reflect the changes including a new section on "Individual components" followed by the existing part under the new defined section "Levels of software organization".

Reviewer comments: Regarding inputs, it would be helpful to describe more specifically what inputs are required and how they are input into the system. The article is very vague on this point, indicating only that one would need and "advanced user" to understand these things. The article should indicate what inputs are provided through files that are read by the system and how these files must be structured. Also, which inputs are provided through the GUI, and at what point in the process does this occur? There is a paragraph on importing data, but this paragraph does not indicate specifically what data are needed by the system, only what kind of systems have been used to generate input data. If I wanted to use a different system (or even one of those listed) what would I need to do to create input files for MultiOptForest?
Authors response: We thank the reviewer for highlighting this issue. We wanted the input data to be as flexible as possible, so that input data could be from any forest simulator. Of course, the optimization problem requires a standardized data format, however this format should be possible to create using data processing techniques. Currently, the data is structured as a table – with example datasets provided in the Code Ocean test data. However, as long as a table can be structures with the indices of "stand", "management style" and "year", the optimization software can construct relevant (user defined) objectives and constraints.

To address this issue, we have reorganized figure 1 to use a “input-process-output“ format. We anticipate that expert users will structure the input data, and frame the objectives and constraints, while normal users can interact with the GUI. Currently, only the user preferences information is provided through the GUI. We are considering options to enable user construction of objectives and constraints, however currently they are accessible to moderately advanced coders.
If you wish to use a different forest simulator – the data currently needs to be structured as it is in the examples, or the code needs to be edited so that the data is modified to reflect that structure. The structure is a table, with indices for the ‘stand’, ‘management style’ and ‘year’ – all of the columns are then variables (e.g. ecosystem service and biodiversity indicators) that describe that specific stand, according to that management style at that specific time period.
We added a paragraph to the “importing data” section saying: Examples for the importing data of the above four forest models are on Code Ocean and can be interactively loaded for the below use case following the step-by-step guide included in the supplementary material and linked on the Readme.md instructions (Getting started).

Reviewer comments: In the middle of the system (somewhere) an (or more likely many...) optimization problem is formulated and solved. The article discusses how different solvers can be used, but, again, what would I have to do if I wanted to use Gurobi - for instance - within MultiOptForest?
Authors response: There are a few optimizations that are conducted to enable the use of the achievement scalarizing function (to estimate the nadir and ideal values of each objective), however at the end, the interactive optimization problem is a single multi-objective optimization function using the achievement scalarizing function with epsilon constraints (with an option to include specific management restrictions). By adjusting the reference points and epsilon constraint values, new Pareto efficient solutions are possible. To adjust the solver, all one needs to do is appropriately call the solver in the code. Of course, with commercial solvers appropriate permissions are required. We have currently coded a selection of solvers to be used (GLP, CPLEX, GUROBI, GLOP). To switch from one to another, we need to call the specific solver that is being used. This was hard coded into the software, however to address this comment we have incorporated a variable that can be adjusted when you call the optimization function. So now, to change solvers, you need to simply identify it when calling the function (currently it is set to a default of CLP).
So now, when starting the GUI, the solver can be identified directly, i.e.:
BES = Opt.OptGUI("NFS",True,"rslt_RCP0_Bavaria_Germany_pause_2_V1.zip",solver="CLP")
We have updated this information in the Readme.md of the software on Code Ocean, to highlight this adjustment. Only open access solvers will work on the Code Ocean system, but we have tested on a system with CPLEX, and this works nicely.

Reviewer comments: On the output side, the article offers three vague sentences about the program's output. I would like to know more about what is included in the output. Does it look something like Figure 3? How might the output look for a different type of forest, perhaps with different management options and different objectives? What if I want more detail about a given Pareto optimal solution? How can I visualize trade-offs among different preference sets? Figure 3 shows results for one set of preferences from the case study, but it does not illustrate how the system was used to assess trade-offs between different

objectives, which, to me, is a key point of multiobjective optimization.
Authors response: Currently, the output of the optimization is something similar to Figure 4 (now after revisions), in addition to a table of the management options selected for each stand, with a column to indicate how much of the stand is managed according to the specific management option (as linear programming is used). This table includes all information produced by the forest simulator, so users can produce a wide range of visualizations for all variables that were simulated in the simulator (and not just those objectives and constraints that were included in the optimization).
Yes, figure 4 does show the results of one case study, and doesn't assess trade-offs between objectives. This wasn't the focus of the optimization approach, nor was this the focus of the software. However, this software could be used to assess trade-offs, if desired. This would require storing a variety of solutions to enable comparisons. To visualize trade-offs between preference sets, you would need to extract solutions for each preference set and compare. Or alternatively, produce values for each objective, and compare these. If the interactive process is quick enough – users will be able to intuitively deduce trade-offs between objectives. However, from our perspective, multi-objective optimization was used to construct preference specific solutions, and not specifically to quantify and evaluate trade-offs.
Overall, the optimization output is primarily a very large table, with the outcomes of the specific output. A variety of graphs, maps and other visualizations are possible. A more detailed explanation we have added to manuscript, see end of section Graphical user interphase. Currently we have coded a selection of time series graphs, maps representing the spatial information of the data (if there is a corresponding geographic data). Currently we do not store the information of multiple solutions – although this is a very good idea and we will incorporate this to the next version of the software. You can store the information through a csv file, or by capturing the images constructed in the software.

Reviewer comments: Major comments: Constraints vs Objectives
The article is not clear on the distinction between constraints and objectives. For example, one item listed as an objective in the section "Formulating objective functions" sounds more like a constraint than an objectives. In particular, "increase the quantity of deadwood in the forest by a specific year, and to ensure that quantity is kept for the rest of the planning horizon" sounds like a constraint to me. Furthermore, the software offers a dozen ways to create an objective but only (apparently) two ways to create a constraint. In the case study (see Table 2), objectives and constraints are mixed together with no indication which items are objectives and which are constraints.
Authors response: The reviewers are somewhat correct that there may be some confusion on the distinction between constraints and objectives. In this case, we are using a multi-objective optimization formulation, with a variety of functions representing the achievement of a wide variety of variables over time, evaluated as specific functions. All functions can be defined as both constraints and objectives – constraints are formulated using the epsilon constraint approach (Haimes et al. 1971) while the objectives are formulated using the achievement scalarizing function (Weizerbicki 1982) . The key difference is that the epsilon constraints are given values that are required to be above (this introduces the possibility of infeasible solutions) while objectives are given targets which the ASF strives to meet as close as possible. Thus, the clear distinction between constraints and objectives is the mathematical formulation applied to the specific function, either the

epsilon constraint (at line 392) or the ASF (lines 389 and 390).
So, when this reviewer suggests that there are “a dozen ways to create an objective”, these objectives can also be interpreted as constraints (with the difference being that a constraint is something that must be met, while the reference point objective is something that is desired).
The additional constraints (e.g. only CCF management on peat as in the Finnish example) are not epsilon constraints, but “hard” constraints on management.
Literature:
Wierzbicki AP: On the completeness and constructiveness of parametric characterizations to vector optimization problems. OR Spektrum. 1986;8:73–87. 10.1007/BF01719738

Reviewer comments: Minor comments:
The article states that: "Forest planning literature often focuses on the production of timber resources, obtaining a relative even-flow of timber over time while minimizing costs of logistics and maximizing timber production." This is not accurate. There is a large literature of forest planning models that address multiple objectives.
Authors response: We have change the corresponding section in the introduction (also according R1). Please see our changed section, saying that “A few centuries ago, forest planning literature often focused on the production of timber resources”. Additionally we highlighted two sentences later, that “Nowadays, forests are instead important and recognized for a wide range of societal demands”

Reviewer comments: The article references two similar packages, desdeo-mcdm and optimLanduse, but it does not elaborate on what the similarities and differences are between MultiOptForest and these other packages. For example, could one use either desdeo-mcdm and optimLanduse instead of MultiOptForest?
Authors response: Desdeo-mcdm – is a industry independent software solutions, likely aspects could be incorporated, whereas our software is adapted to forest management planning problems. optimLanduse used optimLanduse uses an robust multi-objective land-cover composition optimization approach of Knoke et al.(2016)., while MulitOptForest used a multi-objective problem formulation (Miettinen, 1999) in combination with the achievement scalarizing functions approach (Wierzbicki 1986). This difference we have explained now in the introduction section.
Literature:
Knoke, T., Paul, C., Hildebrandt, P., Calvas, B., Castro, L.M., Hartl, F., Dollerer, M., Hamer, U., Windhorst, D., Wiersma, Y.F., Curatola Fernández, G.F., Obermeier, W.A., Adams, J., Breuer, L., Mosandl, R., Beck, E., Weber, M., Stimm, B., Haber, W., Fürst, C., Bendix, J., 2016. Compositional diversity of rehabilitated tropical lands supports multiple ecosystem services and buffers uncertainties. Nature Communications 7. https://doi.org/10.1038/ncomms11877

Reviewer comments: I would like to acknowledge the assistance of Silvana Nobre in developing this review.
Authors response: We appreciate the comments of both reviewers!

Competing Interests: No competing interests were disclosed.

Reviewer Report 06 October 2023

https://doi.org/10.21956/openreseurope.17082.r35231

Can Vatandaslar
Artvin Coruh University, Artvin, Turkey

Manuscript Type

Software tool article

General comments:

The manuscript highlights the importance of decision support tools in multifunctional forest management planning and introduces a software, MultiOptForest, which is capable of analyzing synergies & trade-offs among different objectives and ecosystem services under various land-use scenarios. It also provides a case study to show the capability of the software.

The software is original and it fills an important gap in management-related decision making in forestry that can be often complicated under conflicted objectives and societal demands. Although the authors state that MultiOptForest provides an interactive optimization framework to non-expert users without training in coding, I could not find a graphical user interface when I reviewed the Zenodo repository (i.e., Eyvindson et al. 2023). It is all about codes written in the Python language which can hinder the uptake of the proposed solution, especially considering the coding skills of forest managers and private landowners across the world.

Thus, my suggestion is that the GUI of the software (as shown in the MS) should be made accessible to readers as soon as possible to speed up the uptake process of the software among decision-makers.

My small comments related to specific sections of the MS are below:

Specific comments:

Title

“MultiOptForest: An interactive multi-objective optimization tool for forest planning and scenario analysis”.

The title seems clear and concise.

Abstract

"... multiple forest ecosystem services and biodiversity". According to the latest version of CICES (v5.1), biodiversity is one of the ecosystem services. So, it may be removed from the end of this sentence.

"... impacting the development of forests..." may be revised to "...impacting the natural development of forests..." if the intent is forest growth and yield over time.

Introduction

"... multiple forest ecosystem services and biodiversity, ...". Please see my previous comment for the abstract section.

"... to plan the silvicultural and harvesting decisions..." ...quantify the trade-offs between multiple forest objectives." If MultiOptForest can also analyze the synergies among multiple ecosystem services, the authors should add this term (i.e., synergy) along with trade-offs.

The Pareto optimal solution approach was briefly introduced in the second paragraph. Some references are needed here to inform interested readers.

"Forest planning literature often focuses on the production of timber resources..." I do not agree with the authors. It might be somehow true before the 1990s (as shown in the work cited by the authors (Johnson & Scheurman, 1977), but today contemporary forest management plans mostly deal with multi-functionality, focusing on both economic, ecological, and sociocultural aspects of sustainability. I would suggest a broader literature search for the most recent studies across the world beyond their own work (i.e., Blattert et al. 2022).

"...this software can be used in a variety of forest planning cases, including tactical, strategic, and operational planning cases". How about scales? Which scale does work well with this software? For example, can it be easily run at the national scale (larger problems)? The spatial aspect and potential limitations should be stated in the manuscript. (I have noticed that the authors mentioned the scales in the discussion. A brief description would also be okay in the Intro).

Software description

While incorporating stand ID, time period, and management regime into the software is clearly explained, integrating the provision of ecosystem services seems vague.

GUI

I could not find the GUI in the zip file the authors provided through Zenodo.

Use case

In Table 2, Maximizing wood production is objective, while even-flow harvest is a constraint. If so, they should be separated into two cells.

Discussion and Conclusion

These sections seem appropriate.

The English grammar and academic language of the manuscript is good as a whole.

Best regards.

Is the rationale for developing the new software tool clearly explained?
Yes

Is the description of the software tool technically sound?
Yes

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?
Partly

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?
Partly

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: My areas of expertise are forest management planning, ecosystem services, and modeling. I do not have enough skills to assess the codes written in the Python language.

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard.

Author Response 02 Dec 2024

Kyle Eyvindson

Authors response:
We appreciate the constructive review. To respond to the suggestions and comments, we respond to the reviewer directly.
Reviewer comments:
The manuscript highlights the importance of decision support tools in multifunctional forest management planning and introduces a software, MultiOptForest, which is capable of analyzing synergies & trade-offs among different objectives and ecosystem services under various land-use scenarios. It also provides a case study to show the capability of the software. The software is original and it fills an important gap in management-related decision making in forestry that can be often complicated under conflicted objectives and societal demands. Although the authors state that MultiOptForest provides an interactive optimization framework to non-expert users without training in coding, I could not find a graphical user interface when I reviewed the Zenodo repository (i.e., Eyvindson et al. 2023). It is all about codes written in the Python language which can hinder the uptake of the proposed solution, especially considering the coding skills of forest managers and private

landowners across the world.
Thus, my suggestion is that the GUI of the software (as shown in the MS) should be made accessible to readers as soon as possible to speed up the uptake process of the software among decision-makers.

Authors response:
We completely agree that the use of the GUI needs to be accessible to the users. We have tried to do this with the Code Ocean framework (https://codeocean.com/capsule/2689439/tree/v1) – as it allows for temporary (and free for academics) use of cloud computing. This allows users to test the tool without any software installations on their local computer. We now have created a step-by-step user guideline as supplementary material (but also linked to the Readme.md) allowing users to get started with the tool on how to connect with the Code Ocean framework, launching the tool, testing and running the optimization with the GUI. This should provide a good basis for accademics/forest planners. Of course, a more advanced software implementation would require additional programming efforts which is unfortunately not doable at this stage.

Reviewer comments:
My small comments related to specific sections of the MS are below:
Specific comments:
Title
"MultiOptForest: An interactive multi-objective optimization tool for forest planning and scenario analysis". The title seems clear and concise.

Authors response:
Thanks.

Reviewer comments:
Abstract
"... multiple forest ecosystem services and biodiversity". According to the latest version of CICES (v5.1), biodiversity is one of the ecosystem services. So, it may be removed from the end of this sentence.

Authors response:
We kindly disagree with the reviewer. Looking at the CICES (V5) classification https://cices.eu/cices-structure/ the hierarchical structure of the earlier version (CICES V4.3) has been retained in V5.1. At the highest level in the classification, services are grouped according to three Sections that relate to whether the contributions to human well-being support:
a) the provisioning
of material and energy needs,
b) regulation and maintenance of the environment for humans, or
c) the non-material characteristics of ecosystems that affect physical and mental states of people, that is their cultural significance.
Thus, we prefer to talk about ecosystem services "and biodiversity" in the abstract (and the manuscript), and aim to follow the classification of https://foresteurope.org/ecosystem-services/. According to this classification, biodiversity is seen as: "the variety of all life on

earth – plays a key role in the structural set-up of ecosystems, which is essential to maintaining basic ecosystem processes and supporting ecosystem functions."

Reviewer comments:
"... impacting the development of forests..." may be revised to "...impacting the natural development of forests..." if the intent is forest growth and yield over time.

Authors response:
We changed it to ..."impacting the future development of forests", since it can cover natural aspects as well as management aspects (planting, tending, harvesting, etc.).

Reviewer comments:
Introduction
"... multiple forest ecosystem services and biodiversity, ...". Please see my previous comment for the abstract section.

Authors response:
Please see our comment above.

Reviewer comments:
"... to plan the silvicultural and harvesting decisions..." ...quantify the trade-offs between multiple forest objectives." If MultiOptForest can also analyze the synergies among multiple ecosystem services, the authors should add this term (i.e., synergy) along with trade-offs.

Authors response:
We have changed it to: "This has led to a rising demand for tools to quantify the conflicts between multiple forest objectives. These tools should be able to investigate the trade-offs and synergies between objectives, ..."

Reviewer comments:
The Pareto optimal solution approach was briefly introduced in the second paragraph. Some references are needed here to inform interested readers.

Authors response:
We have included the following references to this paragraph:
Borges, J.G., Garcia-Gonzalo, J., Bushenkov, V., McDill, M.E., Marques, S., Oliveira, M.M., 2014. Addressing Multicriteria Forest Management With Pareto Frontier Methods: An Application in Portugal. Forest Science 60, 63-72. http://dx.doi.org/10.5849/forsci.12-100
Díaz-Yáñez, O., Pukkala, T., Packalen, P., Lexer, M.J., Peltola, H., 2020. Multi-objective forestry increases the production of ecosystem services. Forestry: An International Journal of Forest Research. https://doi.org/10.1093/forestry/cpaa041
Mazziotta, A., Podkopaev, D., Triviño, M., Miettinen, K., Pohjanmies, T., Mönkkönen, M., 2017. Quantifying and resolving conservation conflicts in forest landscapes via multiobjective optimization. https://www.silvafennica.fi/article/1778

Reviewer comments:

“Forest planning literature often focuses on the production of timber resources…” I do not agree with the authors. It might be somehow true before the 1990s (as shown in the work cited by the authors (Johnson & Scheurman, 1977), but today contemporary forest management plans mostly deal with multi-functionality, focusing on both economic, ecological, and sociocultural aspects of sustainability. I would suggest a broader literature search for the most recent studies across the world beyond their own work (i.e., Blattert et al. 2022).

Authors response:
Thanks for the comment. We have re-phrased the corresponding section; highlighting, that “A few centuries ago, forest planning literature often focused on the production of timber” (see paragraph three).
We also highlighted that “Nowadays, forest are instead important and recognized for a wide range of societal demands (LIT-1), why there is a need to shift towards a flexible approach accounting for those multi-objectives (LIT-2).” Those statement we complemented by following Literature
LIT-1:
MEA, 2005. Millenium Ecosystem Assessment - Ecosystem and Human Well-being: Snthesis. Island Press, Washington, DC.
Pascual, U., Balvanera, P., Díaz, S., Pataki, G., Roth, E., Stenseke, M., Watson, R.T., Başak Dessane, E., Islar, M., Kelemen, E., Maris, V., Quaas, M., Subramanian, S.M., Wittmer, H., Adlan, A., Ahn, S., Al-Hafedh, Y.S., Amankwah, E., Asah, S.T., Berry, P., Bilgin, A., Breslow, S.J., Bullock, C., Cáceres, D., Daly-Hassen, H., Figueroa, E., Golden, C.D., Gómez-Baggethun, E., González-Jiménez, D., Houdet, J., Keune, H., Kumar, R., Ma, K., May, P.H., Mead, A., O’Farrell, P., Pandit, R., Pengue, W., Pichis-Madruga, R., Popa, F., Preston, S., Pacheco-Balanza, D., Saarikoski, H., Strassburg, B.B., van den Belt, M., Verma, M., Wickson, F., Yagi, N., 2017. Valuing nature’s contributions to people: the IPBES approach. Current Opinion in Environmental Sustainability 26-27, 7-16. https://doi.org/10.1016/j.cosust.2016.12.006
Winkel, G., Lovrić, M., Muys, B., Katila, P., Lundhede, T., Pecurul, M., Pettenella, D., Pipart, N., Plieninger, T., Prokofieva, I., Parra, C., Pülzl, H., Roitsch, D., Roux, J.-L., Thorsen, B.J., Tyrväinen, L., Torralba, M., Vacik, H., Weiss, G., Wunder, S., 2022. Governing Europe's forests for multiple ecosystem services: Opportunities, challenges, and policy options. Forest Policy and Economics 145, 102849. https://doi.org/10.1016/j.forpol.2022.102849
LIT-2:
Augustynczik, A.L.D., Gutsch, M., Basile, M., Suckow, F., Lasch, P., Yousefpour, R., Hanewinkel, M., 2020. Socially optimal forest management and biodiversity conservation in temperate forests under climate change. Ecological Economics 169, 106504. https://doi.org/10.1016/j.ecolecon.2019.106504¨
Eggers, J., Lundström, J., Snäll, T., Öhman, K., 2022. Balancing wood production and biodiversity in intensively managed boreal forest. Scandinavian Journal of Forest Research 37, 213-225. https://doi.org/10.1080/02827581.2022.2066170
Uhde, B., Heinrichs, S., Stiehl, C.R., Ammer, C., Müller-Using, B., Knoke, T., 2017. Bringing ecosystem services into forest planning – Can we optimize the composition of Chilean forests based on expert knowledge? Forest Ecology and Management 404, 126-140. https://doi.org/10.1016/j.foreco.2017.08.021

Reviewer comments:

"...this software can be used in a variety of forest planning cases, including tactical, strategic, and operational planning cases". How about scales? Which scale does work well with this software? For example, can it be easily run at the national scale (larger problems)? The spatial aspect and potential limitations should be stated in the manuscript. (I have noticed that the authors mentioned the scales in the discussion. A brief description would also be okay in the Intro).

Authors response:
We appreciate this comment, the issue of scale can be related to the time taken in the data processing and optimization. We have applied the software with several very large problems and were able to manage successfully. As the reviewer noted, the scale aspect in the discussion (second last paragraph). However, we briefly specified the flexibility of the tool (applicable to large and small scale) also at the end of the intro saying: "As a flexible approach to constructing multi-objective optimization formulations, this software can be used in a variety of forest planning cases, including large scale strategic planning cases (e.g. national forest inventory plot level) as well as small scale tactical and operational planning cases (e.g. individual forest stands of a management unit)."
The software works well for both scales. However, computation time and complexity depend on the number of the number of individual optimized entities (forests plots or stands), which we have added to the discussion.

Reviewer comments:
Software description
While incorporating stand ID, time period, and management regime into the software is clearly explained, integrating the provision of ecosystem services seems vague.

Authors response:
We improved the description of indexed input data to the provided ecosystem services and biodiversity and wrote: "... columns for indicators represent the state of the forest and of the ecosystem service and biodiversity. Those indicators are simulated for each entity (forest stand or plot level) under alternative managements."

Reviewer comments:
GUI
I could not find the GUI in the zip file the authors provided through Zenodo.

Authors response:
The GUI is not found on the Zenodo, the Zenodo is only a repository of the python code (as the reviewer has correctly noted). To enable readers to interact with the code, we have loaded the code as a project in the virtual environment on Code Ocean, where those interested can test and explore the approach.
https://codeocean.com/capsule/2689439/tree/v1 .
We generated a step-by-step guide illustrating with screen shots how the tool can virtually launched via CodeOcean, providing as such an illustrative example how user can get familiar with the tool and explore pre-defined cases/optimization problems, including: how to start the tool on Code Ocean (start editing), how to Launch Jupiter notebook, selecting one of the country example, and run the Jupyter Notebook, which loads the GUI where

constraints/objective can be set.
The step-by-step user guide was further included in the ReadMe file, showing now two sub-section under "Getting started": Online on Code Ocean, and "On local system"

Reviewer comments:
Use case
In Table 2, Maximizing wood production is objective, while even-flow harvest is a constraint. If so, they should be separated into two cells.

Authors response:
We highlighted, which aspects are addressed as objectives, and which ones as constraints. Wood production and Bioenergy where implemented as objective that maximize the minimum indicator values (harvested roundwood or biomass) over all simulated years, which results in an even-flow of wood or biomass production. This we have further explained in the text and table 2.

Reviewer comments:
Discussion and Conclusion
These sections seem appropriate.

Authors response:
Thanks

Reviewer comments:
The English grammar and academic language of the manuscript is good as a whole.

Authors response:
Thanks

Competing Interests: No competing interests were disclosed.