

Decisi-o-rama

An open-source Python library for multi-attribute value/utility decision analysis

Chacon Hurtado, J.; Scholten, L.

DOI

[10.1016/j.envsoft.2020.104890](https://doi.org/10.1016/j.envsoft.2020.104890)

Publication date

2020

Document Version

Final published version

Published in

Environmental Modelling & Software

Citation (APA)

Chacon Hurtado, J., & Scholten, L. (2020). Decisi-o-rama: An open-source Python library for multi-attribute value/utility decision analysis. *Environmental Modelling & Software*, 135, 1-12. Article 104890. <https://doi.org/10.1016/j.envsoft.2020.104890>

Important note

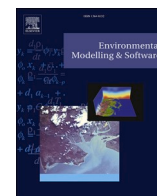
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy

Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.



Decisi-o-rama: An open-source Python library for multi-attribute value/utility decision analysis

J.C. Chacon-Hurtado^{*}, L. Scholten

Delft University of Technology, Department of Water Management, Stevinweg 1, 2628CN, Delft, the Netherlands

ARTICLE INFO

Keywords:

MAUT/MAVT

Portfolio decision analysis

Multi-criteria decision analysis

Python

ABSTRACT

Environmental decisions are complex as they are multi-dimensional, highly interdisciplinary and not only involve multiple stakeholders with conflicting objectives, but also many possible alternatives with uncertain consequences. The difficulty lies in making trade-offs between tough value trade-offs on the one hand while appreciating uncertain impacts of alternatives on the other. To support decisions tackling such problems, a combination of multi-criteria decision analysis (MCDA) and environmental models is promising yet limited by the available MCDA software. Here, we present Decisi-o-rama, an open-source *Python* MCDA library for single and sets (portfolios) of alternatives in the context of multi-attribute value/utility theory (MAUT/MAVT). Its development was driven by four aspirations that are crucial for usability in the context of environmental decision-making: (1) interoperability, (2) uncertainty-awareness, (3) computational efficiency, and (4) integration with portfolio decisions. The results indicate that these aspirations are met, thus facilitating the adoption of MCDA methods by environmental researchers and practitioners.

1. Introduction

Environmental decisions are often complex as there are usually many courses of action, often conflicting, and uncertain. The reason for this complexity lies in how the utilisation of natural resources has sociopolitical, environmental, economic, and even ethical implications. At the time, these decisions are often supported by uncertain and often incomparable data that need to be synthesized. To support decision-makers, (multi-criteria) decision-analysis provides a framework to systematically evaluate decisions, providing transparency, traceability and reproducibility in the process. As a result, there has been an increasing interest in MCDA methods for environmental decision-making (Kiker et al., 2005; Huang et al., 2011; Linkov and Moberg, 2017; Esmail and Geneletti, 2018; Ortiz et al., 2018) encompassing: decision support (Langhans et al., 2014; Lahtinen et al., 2017; Haag et al., 2019; Kuemmerlen et al., 2019; Marttunen et al., 2019), constructing ecological indicators (Convertino et al., 2013; Cinelli et al., 2014; Langhans et al., 2014; Zheng et al., 2016; Blatter et al., 2017), and the support of planning activities (Joubert et al., 2003; Gómez-Delgado and Tarantola, 2006; Hajkowicz, 2007; Calizaya et al., 2010; Huang et al., 2011; Sa-nguanduan and Nitivattananon, 2011; Lienert et al., 2014; Scholten et al., 2015; Scholten et al., 2017).

Decision analysis aims to formalise “common sense for decision problems which are too complex for informal use of common sense” [30, p.806]. Decisions can become complex when there are multiple decision makers with different views of the issues that need addressing, conflicting goals and manifold possible courses of action, uncertainty about outcomes of different actions, unfamiliarity especially in case of one-off, strategic decisions, high interdisciplinarity, and also different subjective valuation of outcomes and risks. Decision-making is further complicated by the overall goals that are considered to compare alternative courses of action often being incommensurate, and high-dimensional. In this context, recurrent decisions with lower stakes and well-defined objectives are generally easier than one-off decisions with high stakes in complex decision contexts where the objectives need to be identified jointly with the decision-makers in the first place (Keeney, 1982; Bureš et al., 2019). Environmental decisions are often of the latter type.

To account for these complex and inter-related processes in a formal context, decision analysis methods that combine problem structuring and multi-criteria decision analysis (MCDA) models are often used (Marttunen et al., 2017). These methods support the decision-making process by structuring the decision problem, systematising the objectives and assessing their fulfilment and measurable attributes, given a set of alternatives (individual actions or ‘portfolios’ of actions) (Reichert

^{*} Corresponding author.

E-mail addresses: j.chaconhurtado@tudelft.nl (J.C. Chacon-Hurtado), l.scholten@tudelft.nl (L. Scholten).

et al., 2013). This allows to compare among alternatives, thus supporting the decision-makers in exploring the decision space and model results (Voinov and Bousquet, 2010; Voinov et al., 2016). As a result, decisions become more transparent, traceable and reproducible.

MCDA methods are generally classified with respect to how goal attainment is defined: (1) overall value (score or rank), (2) goal, reference or aspiration level, and (3) outranking (Belton and Stewart, 2002; Mendoza and Martins, 2006). Summarising value methods indicate the preference of an alternative (or portfolios of alternatives) relative to others based on a numerical score; these methods include the Multi-Attribute Utility and Value approaches (MAUT and MAVT, respectively) (Eisenführ et al., 2010). Aspiration level methods assess the solutions based on the level of attainment of a set of goals as met or not; an example of this is goal programming (Jones and Tamiz, 2016). Outranking methods use a pair-wise comparison of alternatives to identify a ranking of preferences; examples are ELECTRE (Govindan and Jepsen, 2016) and PROMETHEE (Behzadian et al., 2010) family methods. For a more detailed overview of MCDA and related methods see Figueira et al. (2006).

MCDA models come into play once the decision problem is structured. These imply that: (1) there is clarity about who takes part in the decision, (2) what objectives and criteria are to be considered, and (3) which alternatives are feasible (Mendoza and Martins, 2006). Assuming a summarising value approach, it is then necessary to understand and quantify the impacts of the alternatives on the attributes (indicators to measure the outcomes of alternatives regarding the objectives). The impact of different alternatives on the attributes are obtained from the assumed cause-effect relationship between alternatives and anticipated consequences on the attributes. This can be obtained based on conceptual or mathematical models that make these assumed relationships explicit, or from estimates obtained directly from data or expert knowledge. Here, we use the term ‘assessment model’, to describe the mathematical models that are used to map the (portfolios of) alternatives to the expected outcomes on the attributes.

Next, MCDA models require a preference model to appraise the (portfolios of) alternatives, based on the perceived valuation of the anticipated outcomes and their trade-offs. This preference model should consider the different perspectives of the stakeholders, trade-offs among different competing objectives, risk attitudes, and ambiguity attitudes of the decision-makers (Keeney, 1982; Reichert et al., 2013; Scholten et al., 2015).

In addressing a complex decision problem, we agree with (Belton and Stewart, 2002) in that the purpose of an MCDA model is to provide a focus for discussion, not to prescribe the solution. Therefore, MCDA models are useful to learn about trade-offs among alternatives, and to formulate (‘construct’) decision-maker preferences. Unlike optimisation-driven approaches, the purpose of MCDA models is not to determine a normatively ‘best strategy’ to choose, given a set of conflicting objectives. As decision-making involves judgment and valuation, subjectivity cannot be avoided and the responsibility for the decision and its consequences remain with the decision-maker(s).

Generally, MCDA methods are used to evaluate among different alternatives. However, in many cases the actions to be taken go beyond a single alternative (e.g. which project to conduct), seeking to identify a set of potential alternatives (portfolio) to implement (e.g. a combination of several projects (Salo et al., 2006)). The number and combination of alternatives in a portfolio is subject to constraints, such as the available budget or personnel (Gustafsson and Salo, 2005; Liesjö et al., 2008). Also, contingencies and interactions matter regarding which actions can be combined and whether the outcomes are independent of each other (i.e. add up) or not (e.g. economies of scale, upstream-downstream interactions). Portfolio decision analysis (PDA) integrates portfolio theory and multi-attribute value models (Kleinmuntz, 2007; Salo et al., 2011) to explore optimal deterministic portfolios, or robust portfolios when considering the uncertainty in the decision maker preferences and alternative performance (Liesjö et al., 2007, 2008). Applications of PDA

in MCDA are reported in e.g. Salo et al. (2006), Kleinmuntz (2007), Mild and Sahlo (2009), Convertino et al. (2013), Lahtinen et al. (2017).

To navigate these decision spaces, it is necessary to have tools that help in exploring the consequences of different individual or portfolio alternatives along with their uncertainties. Therefore, it is necessary that modern decision-analysis tools are able to accommodate uncertainty and sensitivity analysis (French, 2003; Gómez-Delgado and Tarantola, 2006; Durbach and Stewart, 2011; Scholten et al., 2015; Esmail and Geneletti, 2018). Conventionally, uncertainty and sensitivity analysis require considerable computational resources for simulation and model evaluation over larger samples of model parameters and inputs, as analytical solutions are either too simplified, unavailable, or too restrictive regarding the characteristics of each particular model. For example, one of us conducted extensive uncertainty and sensitivity analysis using an earlier version of the *Utility* (R) package (Reichert et al., 2013), for which it took days to compute the results for a real-world decision problem on a high-performance computing cluster (Scholten et al., 2015, 2017). That makes on-the-fly exploratory decision modelling with stakeholders unviable. Ideally, decision tools should be computationally efficient (to produce results in a short time), and generic enough to be integrated together with other model exploration tools such as optimisation routines and scenario modelling.

To summarise the identified MCDA modelling needs, we need software or software libraries (ideally open-source) that are (1) extensible and interoperable with complementary modelling and visualisation tools, that (2) support uncertainty-aware MCDA including uncertainty in outcomes and preferences, while also being (3) computationally efficient to be used on-the-fly with decision makers and (4) for portfolio decisions. First, extensibility and interoperability allows to accommodate problem-specific process models, integration into larger decision-support frameworks, and use complementary model analysis tools (such as optimisation and visualisation). Second, uncertainty-aware decision models provide the decision-makers with information about the magnitude and sources of the (quantifiable) uncertainty in the model. Third, a computationally efficient model allows modelling and interaction with decision-makers on-the-fly, providing direct feedback on the consequences of the alternatives or portfolios, and preference model, supporting discussion. Fourth, by including the possibility to add portfolio decisions, it is possible to support realistic MCDA.

To our surprise, we were not able to find any suitable software among the plethora of tools available to support MCDA that was able to do so, let alone for our target method MAUT/MAVT for portfolio decisions, see reviews in (Weistroffer and Li, 2016; Mustajoki and Marttunen, 2017; International Society on MCDM, 2020; Cinelli et al., 2020). We found that most of these tools are either embedded into closed-source decision-support platforms, built over particular data environments (not without a reason), or as commercial software, making it difficult to extend and adapt these to specific needs neither in practice nor research. There are a few notable exceptions, including: *MCDA* (Bigaret et al., 2017) (R, Java), and above-mentioned *Utility* (Reichert et al., 2013) (R). The latter do not, however, provide an alternative valuation framework, and lack the built-in capability to conduct portfolio decision analyses. Lastly, given the ubiquity of use of Python for mathematical modelling and scientific computing, it is surprising that no native MCDA libraries are available for MAVT, MAUT, and PDA. Consequently, we see value in developing a decision analysis tool that fulfils the identified needs and thus: (1) is extensible and interoperable, (2) uncertainty-aware, (3) computationally efficient, and (4) able to integrate portfolio decisions.

In this paper, we present the development and testing of *Decisi-orama*, an open-source *Python* library for MCDA on single-alternatives and portfolios, in the context of MAVT/MAUT. The name *Decisi-orama* uses the suffix ‘orama’ as in panorama, derived from ancient Greek “hórama”, meaning “view” or “spectacle”. The spelling ‘-o-rama’ is popularly used to stress the latter meaning thus “*decision show*” or “*decision spectacle*”, as we aim to provide a platform to expose the story

behind a decision by making it transparent, logic, understandable and for everyone to see. Decisi-o-rama provides an interoperable, efficient and flexible framework to support MCDA using the summarising-value type preference modelling for uncertain decision analysis problems. This paper is organised as follows: First, a literature review of MAVT/MAUT, and the current preference modelling tools is presented. Second, the main features of Decisi-o-rama are presented, exposing the concepts behind its functionality. Third, Decisi-o-rama is used to solve two large synthetic problems to test its performance. Fourth, a portfolio decision analysis problem (as presented in (Lahtinen et al., 2017)) is solved, where additional features of Decisi-o-rama are showcased. Lastly, conclusions, recommendations and future directions are presented.

2. Multi-criteria valuation models

Multi-attribute decision analysis models (including MAVT and MAUT, and referred to only as MAUT from this point on as common in the literature while noting that only the former warrants measurable value functions (Dyer and Sarin, 1979)), next to many integrated assessment models, aim to estimate the utility (or attainment) of an objective, given a set of hierarchically structured attributes (or criteria) (Nardo et al., 2008; Singh et al., 2012; Scholten et al., 2017). MAUT implies that a score can be constructed by aggregating different attributes, which require the construction (elicitation) of a preference model (Keeney and Howard, 2002; Cinelli et al., 2014). Among its more interesting features are its conceptual simplicity and its suitability to include risky choices (Dyer, 2005; Eisenführ et al., 2010).

Preference models following multi-attribute value theory are based on three main elements, including an objectives hierarchy, the assessment of the marginal utilities (or values), and trade-offs among its different objectives. The objectives hierarchy makes a top-down assessment, where the overall objective of the decision is broken down into intermediate objectives, which can be further disaggregated either into other intermediate objectives, or into lowest-level objectives for each of which measurable (or estimable) attributes are defined. It is important that only fundamental objectives are included in the hierarchy. Further reading regarding the formulation of objectives and structuring of objective hierarchies can be found in Eisenführ et al. (2010).

Once the objective hierarchy is defined, the marginal valuation functions over the attributes and the trade-offs among the attributes and objectives are to be elicited. The valuation functions either include the risk preferences of the decision-makers (utility functions) or not (value functions), regarding the attributes and objectives. The trade-offs are elicited by understanding what are the desirable trade-offs (often expressed as importance weights) among the attributes and how these should be aggregated to represent these trade-offs (value or utility 'aggregation function'). This elicitation process yields uncertain parameters of the preference model, as a consequence of the conceptual simplicity of the preference models, preference instability (e.g. Lienert et al. (2016)) and limited interaction with the decision-makers, among other reasons (Scholten et al., 2015).

To summarise, we can represent the components of portfolio decisions in the MAVT/MAUT as presented in Fig. 1. These components also represent the type of problems that Decisi-o-rama addresses. Starting from the top, it is possible to identify the preference model (orange box), the assessment model (blue box) and the alternative portfolios (white box). In the preference model, G represents the overall objective, $O1$ and $O2$ are intermediate objectives, with utilities (values) of $O1$ constructed by aggregating the attributes $AT1$ and $AT2$ and $O2$ being quantified by $AT3$. The overall objective is defined by aggregating $O1$ and $O2$. Continuing, the assessment model permits mapping the consequences of each portfolio (P) of alternatives to attributes outcomes and respective valuation in light of the given preference model. Finally, the alternatives (A) are defined as either single actions, or a collection of actions (portfolios) that are selected (= 1) or not (= 0) as candidates in

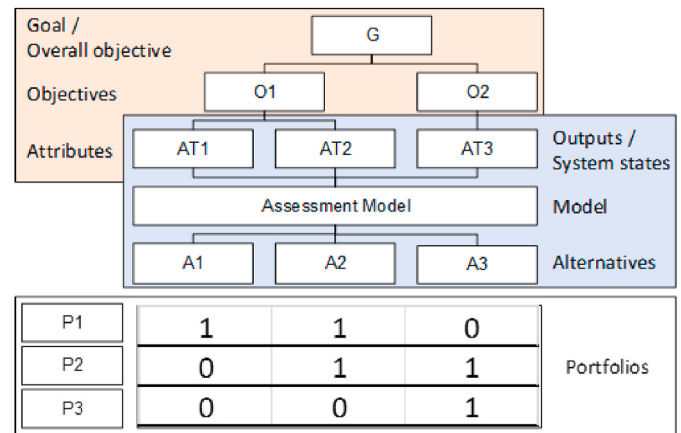


Fig. 1. Representation of the components of portfolio decisions in the MAVT/MAUT context. Here portfolios (sets of actions) are represented in the bottom box. The assessment model is defined by the users, and maps the actions into the attributes of the preference model (middle box). The attributes are hierarchically aggregated in the preference model towards the overall objective (upper box).

the MCDA model.

2.1. Constructing MAVT and MAUT models

Once the objectives and their hierarchy are defined a MAUT preference model needs to be specified to value the alternatives based on the assessment model outputs with regard to the objectives.

2.1.1. Valuating objectives - marginal valuation functions

Valuating an objective in the objectives hierarchy requires defining a value or a utility function. The value of the objective represents the subjective desirability of its level of attainment under certainty. The utility function combines the risk attitude of the decision-maker and the subjective desirability of attaining some level on a particular attribute or objective with quantifiable probability ('risk') (Dyer, 2005). For MCDA with MAUT, utility preferences are modeled through Expected Utility Theory (EUT) (Von Neumann and Morgenstern, 2007).

In EUT, the risk preferences are reflected in von Neumann and Morgenstern-type utility functions obtained from elicitation of preferences over lotteries, according to which the decision-makers are either risk-averse, risk-seeking or risk-neutral. This utility is characterised by relating the value of attributes and utilities to monotonic mappings, in which the utility is absolute, assuming a rational decision (Von Neumann and Morgenstern, 2007). To model the utility of an objective, several approximations have been used in the context of EUT. Common models of utility are the exponential and power model, which correspond to a constant relative and absolute risk aversion respectively. Both approximations use a single parameter (r or p) to indicate the risk-attitude. In Fig. 2 the exponential and power utilities are presented for different risk attitudes, where convexity ($r > 0$ and $p < 1$) represents risk aversion and concavity ($r < 0$ and $p > 1$) represents risk seeking.

2.1.2. Aggregating objectives - overall valuation function

Once the utilities for attributes and objectives are assessed, the utilities corresponding to the attribute levels are aggregated in the intermediate objectives, and later aggregated towards the overall objective. This aggregation requires the definition of the relative importance weights of each attribute (or objective) and the aggregation function. The relative importance of each of the attributes (or objectives) is represented by its weight. The aggregation function determines how to compute an overall score across the individual attributes, representing the trade-offs among attributes (and objectives).

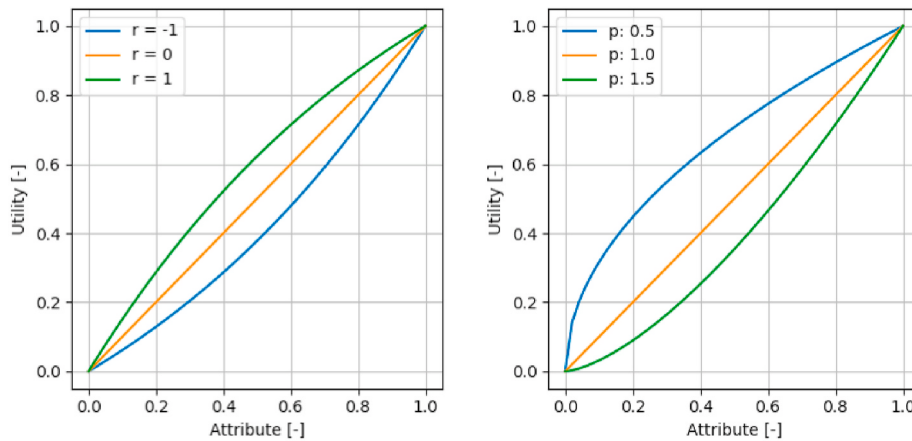


Fig. 2. Exponential and Power utility functions. Positive curvature (r) in the exponential utility and exponent (p) less than 1.0 in the power utility represent risk-averse attitudes. Values of $r = 0$ and $p = 1$ represent risk neutrality. Negative r and p larger than 1.0 represent risk seeking attitudes.

To explore the consequences of different aggregation functions and their impact, we consider aggregation models that represent different preferences regarding compensation between objectives. On the one hand, compensatory models are defined by the existence of a trade-off between the different attributes to be integrated; in other words, a loss in value due to changes in one attribute can be compensated by the gains in another. On the other hand, non-compensatory aggregation models are used to indicate that it is not possible to compensate the losses in a given attribute by improving another. An example of compensatory attributes is revenue, in contrast to non-compensatory aggregation such as bio-diversity indicators (Langhans et al., 2014). The mathematical formulation of some of the most used aggregation functions can be found in (Reichert et al., 2013).

In Fig. 3 we present the impact of different aggregation functions and weights on the trade-off between two attributes, as in (Langhans et al., 2014). These figures show the compensation rate for two attributes (represented in the horizontal and vertical axes), while the contour lines indicate iso-aggregated values. If the lines are vertical or horizontal, then there is no compensation between attributes; if there is a gradient, it describes the degree of compensation between them. Here it is possible to see how aggregation functions like the additive (linear) model are fully compensatory (the trade-off is independent of the value of the attributes), while either the maximum and minimum aggregation functions are non-compensatory. Other functions such as the geometric and harmonic mean indicate a varying (non-linear) compensation where the degree of compensation varies with the value on the respective attributes. Depending on the characteristics of the trade-offs, other conventional (e.g. multiplicative utility (Keeney and Howard, 2002)), mixed (Scholten et al., 2015), or unconventional aggregation functions could be used (Reichert et al., 2019).

2.1.3. Alternatives and portfolios

Alternatives and portfolios are the building blocks of any decision analysis system, as the problems are framed to evaluate their preferability. Alternatives are potential actions that will be decided upon in the decision analysis system. In this respect, a portfolio can be seen as a collection of actions that are simultaneously carried out (Salo et al., 2011).

In many decision-analysis problems, the alternatives correspond to a set of potential actions that can be carried out to affect the system. The characteristics of these actions are specific to each problem, and are typically proposed by teams of experts. Once all the potential alternatives are set, portfolios of alternatives can be constructed by combining individual alternatives and their corresponding actions, where a portfolio of potential alternatives is to be selected.

It is necessary to note that the preferability of a given portfolio

cannot be interpreted as the sum (or weighted sum) of its components. The reason for this relates to the non-linearity of the preference and assessment models. This non-linearity can be seen in the form of interactions among alternatives in the resulting attributes (as estimated by the assessment model), utilities in the intermediate objectives of the preference model, and potential constraints (e.g. competition for resources).

3. Decisi-o-rama: features and reference

Decisi-o-rama is an open-source Python library that supports the development of MCDA models for MAVT and MAUT to compare alternatives or portfolios. Decisi-o-rama encompasses the development of the preference models, as well as pre- and post-processing tools to support the MCDA process relying on common Python objects. These features make the integration with attribute (performance) assessment models straight forward, permitting transparent workflows in the implementation of a diverse range of MAVT/MAUT models. In addition, the tool is focused on simplicity and extensibility, allowing the users to implement user-defined components in the framework, without necessarily modifying the source code.

One of the main features of Decisi-o-rama is the recursive execution of the objectives hierarchy. This allows to: (1) execute large models with minimum impact on memory usage, and (2) flexibly evaluating only parts of the objective hierarchy without model redefinition. As a consequence, it is suitable for the implementation of large decision models in conventional hardware configurations, making on-the-fly model exploration in group settings feasible.

The construction of the objectives hierarchy is separated in two instances: (1) the creation of the individual objectives (nodes), and (2) the creation of the hierarchy of objectives. In Decisi-o-rama, nodes in the objectives hierarchy including the attributes, intermediate objectives, and overall objective are conceptualised as objectives, thus simplifying model construction. The creation of the nodes requires defining attributes, weights, value and aggregation functions. In case the node corresponds to an attribute, the aggregation function will not be considered; in the same way, if the node is an objective (intermediate or overall), its value will be calculated from the aggregation of the lower-level objectives.

The attributes are the outcome of an alternative and can be defined either as a fixed value (deterministic), vector (representing samples from a probability distribution function (PDF) in a Monte Carlo experiment), or generators (routines that yield PDF samples instead of the samples themselves, by storing only the generation rule and its parameters, see (Yee and van Rossum, 2001)). These outcomes are the result of the mapping of the anticipated outcomes of the alternatives on the

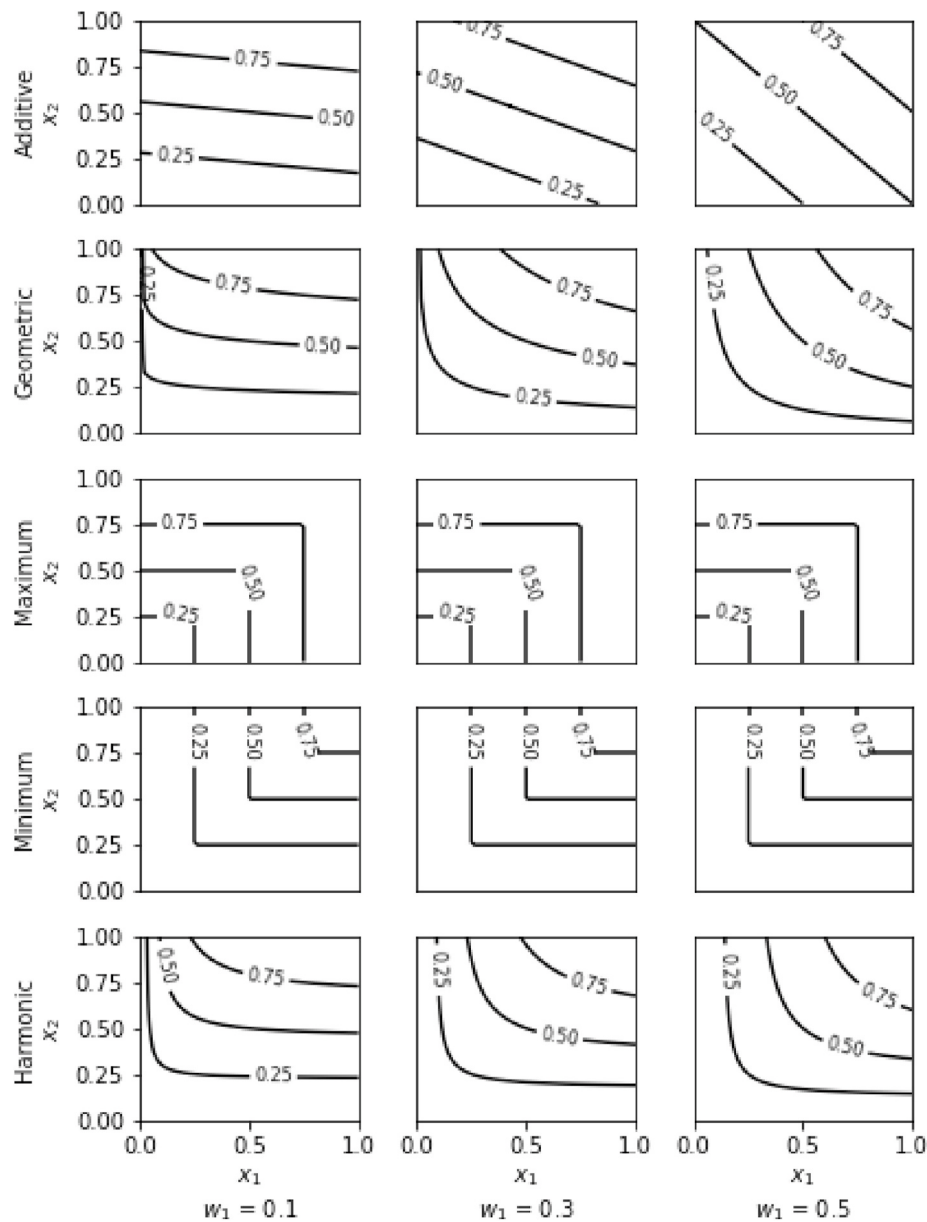


Fig. 3. Aggregated value for different aggregation functions as in (Langhans et al., 2014), using different normalized aggregation weights (w_1) for the attribute x_1 on x-axis, and ($w_2 = 1 - w_1$) for the attribute x_2 on the y-axis.

attributes, which is often carried out using an assessment model (which is problem-dependent and therefore, out of the scope of this document). The objective hierarchy is constructed by mapping the attributes to the lower-level objectives, which again are mapped (aggregated) to the higher-level objectives. Consequently, the execution model will determine the value of each objective as either the value of its attributes, or the aggregated value of its sub-objectives, where the latter overrides the former.

The aggregation functions are defined for each objective, indicating how the lower-level objectives will be aggregated. Weights are also defined for each objective, indicating their relative importance regarding the other objectives at the same level. This is to always ensure a one-to-many relationship between higher-level and lower-level objectives, as there may only be one single weight for each objective in the hierarchy.

A schematic overview of the objective class is presented in Fig. 4. It shows that for each objective in the hierarchy it is required to obtain the (1) Value/Utility (V/U), we require to define the (2) weight and utility

function (if any) for each objective. The value of the objective can be calculated either by (3. a) re-scaling the attributes to the [0–1] interval, of the (4. a) attributes in their natural scale. Alternatively, the objective's value can be calculated from the (3. b) aggregation of the (4. b) value of its sub-objectives with their own defined weights.

3.1. Attribute valuation functions

Currently, exponential and power utility functions (Keeney and Howard, 2002) are implemented in Decisi-o-rama as marginal valuation functions over the individual attributes. The exponential utility function corresponds to a constant absolute risk aversion, while the power utility function to a constant relative risk aversion (Eisenführ et al., 2010). Setting the respective risk attitude parameter, these models simplify to respective marginal value functions, being $r = 0$ for the exponential and $p = 1$ for the power utility function. Both formulations assume that utilities can be measured based on the absolute magnitude of the outcomes, and that probabilities are weighted linearly (i.e. a reduction in

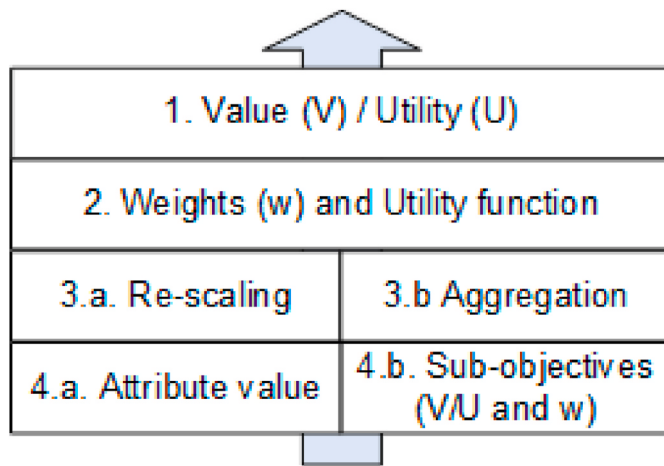


Fig. 4. A conceptual representation of the objective class showing how values/utilities are calculated from attributes (a) or from sub-objectives (b).

risk of 5% is valued equally, no matter if this concerns risk reduction from 50 to 45% or from 5 to 0%).

In addition, Decisi-o-rama supports user-defined utility functions. These functions map a normalized attribute value to a utility value on the range [0, 1]. The function signature receives two parameters: (1) the scalar value in the natural range of the variable (by default [0–1]), and (2) a data structure containing the additional function parameters. These characteristics allows the implementation to be scaleable and generalisable.

3.2. Multi-attribute aggregation models

Decisi-o-rama includes some fundamental aggregation models used to aggregate across multiple attributes. These include: additive (weighted sum), geometric mean (also known as Cobb-Douglas), harmonic mean, power, split power, maximum and minimum aggregation (Langhans et al., 2014). The definition of these basic functions are equivalent to those in the 'R' package *utility* (Reichert et al., 2013). Also, it is possible to include user-defined aggregation functions.

In addition, a meta-function is defined to mix different aggregation functions, as suggested by (Langhans et al., 2013) and (Scholten et al., 2015). This flexibility supports the development of custom aggregation models, which may help in the generalisation and representation of the desirability of objectives (Reichert et al., 2019). As an example, it is possible to define a utility function as the minimum between the additive model and the geometric mean, or make an additive aggregation of the power, split power and geometric means.

3.3. Uncertainty-aware preference and outcome modelling

Next to quantifying the anticipated outcomes of the alternatives on the attributes, one of the main challenges is the elicitation of a preference model. Consequently, this is represented in uncertain estimates of model parameters and aggregation functions. These uncertain parameters propagate through the model, yielding uncertain estimates of the value of objectives at a higher place in the hierarchy, leading to uncertain estimates of the overall valuation of each alternative.

One of the approaches to quantify model uncertainty is Monte Carlo simulation. The implementation of this approach is generally straightforward, yet comes at the cost of considerable processing resources. Therefore, to avoid the overload of a Monte Carlo, we developed an execution model considering three main principles: (1) using generators to sample the PDF's of the parameters (of the utility, aggregation functions and weights) and attribute value (where applicable) for each node, (2) a vectorised execution model of marginal utility and multi-

attribute aggregation functions, and (3) on-demand (lazy) model execution.

First, using random number generators (Yee and van Rossum, 2001) has a significant impact on memory efficiency, as only a representation of the PDF is kept, instead of a sample of random values. This feature allows to dispose intermediate results, keeping only its representation, and making the required PDF samples for the Monte Carlo analysis only available at node execution time. Alternatively, to provide flexibility in the parameterisation of the models (at the cost of memory performance), it is possible to directly pass the random samples of the uncertain inputs or parameters in the analysis.

Second, using a vectorised execution model (van der Walt et al., 2011; Virtanen et al., 2020) for the node operations (attribute valuation and value/utility aggregation) allows considerable speed-up with respect to looped implementations, by making use of modern hardware architecture. This vectorisation is possible as the node operations are relatively simple, which can be integrated into linear algebra frameworks. As a consequence, it is possible to exploit the single-instruction multiple-data processing framework (present in modern processors), allowing to simultaneously compute several instructions within the same CPU cycle.

Third, an lazy execution (Watt, 2006) of the objectives hierarchy permits executing only the required operations to modify only those parts of the problem at the time that are of interest, providing flexibility in the re-evaluation of specific branches. This flexibility permits that nodes are evaluated separately, such as for in-depth analysis of particular parts of the hierarchy, without re-defining the problem. In practical applications, it means that only values or utilities for lower-level objectives are calculated, without executing any of the higher-level objectives, leading to more efficient evaluation of the objectives hierarchy for model exploration.

With these features, it is possible to efficiently execute probabilistic preference models with conventional computing power. This feature also allows to use Decisi-o-rama to explore the impact of different preferences on valuation and outcome computation on the fly. In addition, the library does not require the use of specific data models, making the integration with other applications (such as models for determining the outcome of alternatives, or the implementation of visualisation tools) straight forward.

3.4. Evaluating results

The architecture of Decisi-o-rama separates the execution and the post-processing of results. This allows for different analyses to be carried out without re-executing the model, thus ensuring consistency in the analyses. To support these analyses, we introduce the *Evaluator* class. The *Evaluator* permits to create a container for both: aggregated valuation scores for each alternative, and the performance metrics that summarise the results.

The valuation scores corresponds to the results of the value and utility of the hierarchical objectives, while the performance metrics are the different functions that are used to summarise these results into point values (i.e. mean, median, standard deviation, coefficient of variation). Containing the results into a specific class ensures consistency when applying different performance metrics, and provides a common ground to obtain result analysis tools such as Pareto front identification, and alternative rankings. Similarly to the other components of Decisi-o-rama, the *Evaluator* class also supports user-defined performance metrics.

3.5. Visualising objective hierarchies

Commonly, hierarchical objectives are visualised through trees. This representation provides a logical connection between the hierarchies of nodes, and have been frequently used in different decision-analysis tools to visualise the results and states of a given preference model (Reichert

et al., 2013). These trees serve as a representation of the objective hierarchy, but do not enable to simultaneously highlight model parameters, and results. In addition, using trees to represent a hierarchical model, uses a vast amounts of empty screen (or paper), as the geometric structure its pyramidal, leaving at least half of the available space unused.

To overcome these limitations, we rely on a variant of a sunburst diagram 5. Here, we can visualise several features of the objective hierarchy, including: (1) its structure, (2) weights, and (3) values and/or utilities. The structure of the objective hierarchy is represented in concentric circular wedges, where the internal wedge represents the upper-level objective, while an external wedge represents a lower-level objective. In this direction, the circle in the centre corresponds to the highest-level objective. The weights (w) are represented with the proportional relative area of the wedge (higher weights are represented with wider wedges). The values and utilities are represented using a concentric bar within the wedge that is limited between the 10 (p_{10}) and 90 (p_{90}) percentile, oriented in the anti-clockwise direction, and scaled to the relative width of each wedge. For the particular case of the overall objective, the value of zero is located over the right side of the horizontal axis (in 0° position). The mean value/utility of each objective is represented by the color of the wedge, with lighter colors indicating lower mean values/utilities, while darker colors indicate higher values/utilities. The visualisation in Fig. 5 represents one of the portfolios of the sample problem presented in Fig. 8.

To further understand the proposed visualisation, we will use the example in Fig. 5. Here, we can observe that the average weights for each of the sub-objectives is about 0.25, being represented by wedges of the same size. It can also be seen that *climate* is the sub-objective with the highest variability in value, while *n2* and *p* have the lowest (indicated by smaller orange wedge within the wedge of the respective sub-objective). Regarding mean values, it is possible to observe that *p* and *n2* have the highest values (thus darker color), while *climate* has the lowest among the sub-objectives. Additionally, we see that the overall objective (the circle in the centre), has a mean value approximately of 0.5 (180° counter-clockwise from the horizon on the right), with a p_{10} on 0.46 and p_{90} on 0.55.

3.6. Decisi-o-rama for solving portfolio problems

One of the main features of Decisi-o-rama is the possibility to explore portfolios of alternatives that are linked to a preference model. To this end, it is possible to use Decisi-o-rama in three different ways. First, it is possible to assess the attributes of the lowest-level objectives directly by

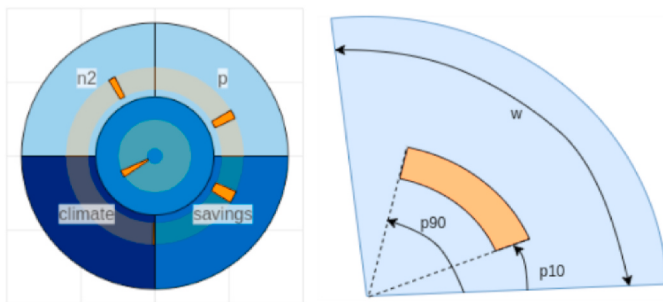


Fig. 5. Visualisation of a hierarchical objective system using the proposed modified sunburst diagram. On the left, the overall objective is represented by the circle in the middle of the plot, while its sub-objectives are represented by concentric wedges. The mean value of the objective is represented by the color of the circle/wedge (darker is higher). The orange bars represent the values of percentile 10 and 90, in the scale 0–1 as indicated in the zoomed figure on the right side (larger uncertainty is represented by a longer orange bar). (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

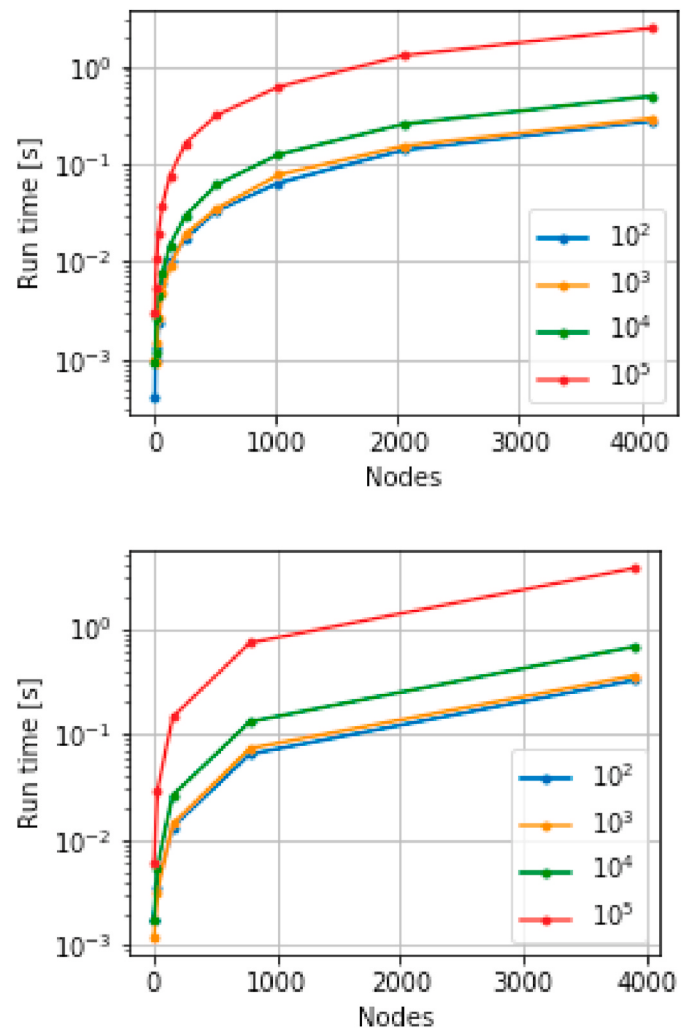


Fig. 6. Computational run time for the evaluation of binary (top) and quinary (bottom) objective hierarchies for differently sized random samples from the distribution of the attributes.

computing the consequences of the outcomes of individual alternatives. Second, if additivity of the alternatives can be assumed, the portfolio valuation model can be fed with the consequences of each action in line with how portfolio decision problems are often framed. Third, it is possible to pass the results of an assessment model directly into the attributes of the preference model. In this context, it is possible to determine not only the portfolio solutions in terms of binary values (present or not), but also as a fraction of a given action or more complex interactions between alternatives (provided by the assessment model).

3.7. Implementation and software availability

Decisi-o-rama has been implemented as an open-source project. As such, we have decided that Decisi-o-rama is published under the MIT license, making it available as it is, without restriction to access, modification or commercialisation. The intention of this licensing is to develop a community of users and collaborators to further advance its development, making MCDA accessible to practice and research.

Decisi-o-rama is available in Python 3. x, making use of only few commonly used libraries, ensuring its sustainability in time, including NumPy (Oliphant, 2006), Numba (Lam et al., 2015) and SciPy (Virtanen et al., 2020). NumPy is used for mathematical and linear algebra operations, Numba is used to interface to the LLVM compiler (Lattner and Adve, 2004), thus supporting performance and scalability, while SciPy

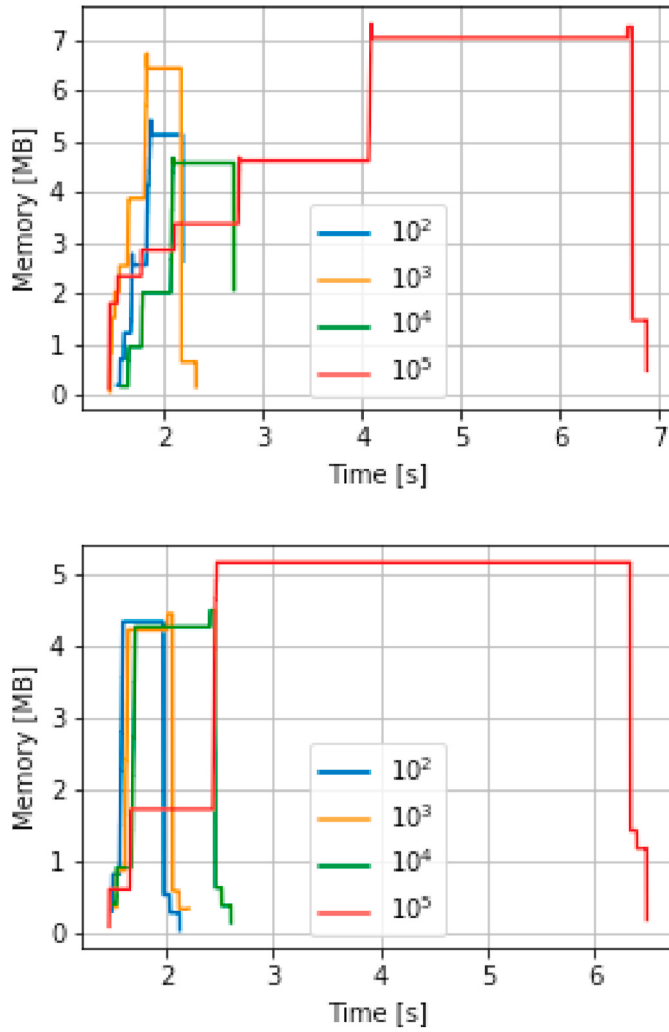


Fig. 7. Memory used in the evaluation of binary (top) and quinary (bottom) objective hierarchies for differently sized random samples from the distribution of attributes (this does not account for the memory use in launching the process).

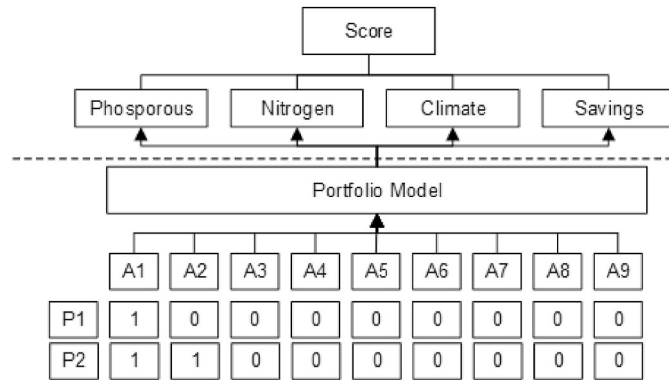


Fig. 8. Summary of the portfolio decision model. A1 ... 9 alternatives, P1 and P2, some potential alternative portfolios.

is limited to its use in calculating percentile statistics in the post-processing (*Evaluator* class).

4. Demonstration

In this section we will explore the performance and usability of Decisi-o-rama. To test its computational performance, we will use it to solve two large synthetic problems with a varying number of levels and nodes in homogeneous objective hierarchies (with the same number of sub-objectives); these types of problems are easily scaleable, and provide information about the capabilities of the tool to solve large problems. Following, the usability of the tool is showcased in modelling the portfolio problem presented in (Lahtinen et al., 2017).

The benchmarks and simulations are carried out in conventional settings. The tests in this implementation were carried out using Python 3.7.7. The benchmark is performed in an Intel Core™ i5-3320M CPU @ 2.60 GHz, with 8 GB of memory using Ubuntu (Linux) 18.04 64-bits. Tests in a Windows environment yielded similar results. The scripts to obtain these results are included in the documentation of Decisi-o-rama.

4.1. Computational performance

To test the computational performance, we are solving a hierarchical aggregation problem using perfect binary (2 sub-objectives) and quinary (5 sub-objectives) trees, for which each additional layer (depth level) in the hierarchy will yield an exponentially growing number of nodes. The analysis is carried out for 2 to 12 depth levels in a binary tree (thus ranging from 3 to 4095 nodes), and for 2 to 6 depth levels in a quinary tree (6–3906 nodes). Although in practice the use of such large objective hierarchies is unrealistic, we present these here as limiting cases for the sake of performance testing.

The evaluation is set to use uniform random values for the attributes in the lowest-level objectives. For the test, we use a varying number of random samples from 100 (10^2) to 100 000 (10^5). In addition, we estimate the marginal utility at each node using an exponential utility function with fixed parameters for the lowest-level objectives, while using an additive aggregation function to estimate the joint utility at the intermediate and overall objectives.

We used common indicators of computational performance, namely wall time (Fig. 6) and memory usage (Fig. 7), obtained on an Intel Core i5 at 2.6 GHz. The results in (Fig. 6) indicate that the computational run time for the experiment is low, even for large problems (100 000 random runs in more than 4000 nodes), with evaluations being computed in the fraction of a second.

To help summarising the information about the performance, two different experiments were carried out. For the run-time analysis (Fig. 6), we calculated the time that it took to run each tree, at different depth levels (from 2 to 12 for the binary, and from 2 to 6 for the quinary tree), for different number of random samples (from 100 to 100 000). For the memory usage (Fig. 7), we use a sequential run, indicating that over the same number of random samples, the trees at different depths are run in the same experiment, yielding a single line over time. It has to be noted that the run-times in Fig. 7 do not perfectly match those on Fig. 6, as it includes the definition of the problem and not only the execution time.

The results show a slightly better performance for the quinary over the binary tree (Figs. 6 and 7). These results are explained by the fact that the binary tree in this experiment is deeper than the quinary tree, thus requiring the allocation of a larger number of stacks before the calculations are executed. Due to the architecture of Python, stacks tend to be rather big (in comparison to other lower-level programming languages) thus accounting for a more noticeable impact on memory use and run-times, than the amount of samples used in the experiment. However, run-time and memory used in computing both of the experiments show an almost negligible impact (below 7 MB) for current-day hardware configurations.

These results show that Decisi-o-rama allows to handle large problems using little resources. The size of the models depends largely on the number of objectives, and not on the number of random samples that are drawn in the Monte Carlo experiment (Fig. 7). This leads to large models being stored with almost negligible memory footprint, making it suitable for analysis on nowadays standard hardware configurations.

4.2. Using Decisi-o-rama in a portfolio decision analysis problem

In this section we use Decisi-o-rama to the portfolio decision analysis problem as presented in Lahtinen et al. (2017). A detailed description is given in the original reference. Here, we will only describe those aspects which are relevant for demonstrating Decisi-o-rama for this decision problem.

The decision problem is to support the selection of a portfolio of alternatives to reduce the water demand in the city of Bass. The overall objective consists not only in the saving of water volume (%), but also considers long-term financial effects (Million Australian Dollars, MAUD), climate change related impacts (Score) and consequences for the local water system (reduction in Phosphorous and Nitrogen Ton/yr). The problem is represented based on six attributes (Table 1):

In this example, the cost and water demand are used as constraints to create portfolios, as a water saving target of 50% should be achieved, given a budget of 45 MAUD. Therefore, the value model is constructed over the remaining attributes in the model. To identify the best portfolio given these 6 objectives, 9 potential alternatives to reduce water demand in the area are considered (Table 2):

The consequences of these alternatives are given in Table 3, assuming a uniform distribution of the attributes within the given ranges in the brackets.

In addition, a series of constraints define which alternatives can be combined into a potential solution portfolio (Table 4):

As a result, the problem can be conceptualised as hierarchical estimation of the overall value based on aggregation of four attributes, see Fig. 8. The alternatives are binary, indicating that they are executed (value of 1), or not (value of 0), to create the vector representing a portfolio. In Fig. 8, we present only two of the possible portfolios for illustration where in the first portfolio (P1) only alternative 1 is considered, and in the other (P2) alternative 1 and 2 are selected in combination.

In the original decision problem, the weights are assumed equal. Here, we add uncertainty to the weights, assuming that these come from the same normal distribution where $w_i \sim N(0.25, 0.05)$. Additionally, the attributes of the alternatives are modeled using a normal distribution within the upper and lower bounds as described in Table 3.

Let us assume that the preferred portfolio is then the one with the highest expected value and smallest uncertainty (and ignore more complex preferences with regard to uncertainty in the outcomes). The optimal set of solutions in a multi-objective optimisation problem are found in the non-dominated (Pareto) set, where each of the solutions is better on at least one objective and at least equal over the other (dominated) solutions. The expected value is calculated using the mean of the distribution, while its uncertainty is expressed by the interquartile range (difference between the percentile 25 and 75, IQR, Fig. 9). To complement the analysis, we add another metric of variation in the form

Table 1
Attributes of the test problem.

Attribute	Description	Symbol	Units
1	Reduction in P release	P	Ton/yr
2	Reduction in N2 release	N2	Ton/yr
3	Climate change impacts	Climate	Score
4	Long-run savings	Savings	MAUD
5	Implementation costs	Cost	MAUD
6	Reduction in water demand	Water	%

Table 2
Alternatives of the test problem.

Alternative	Description
1	Toilets with reduced water consumption
2	Showers and faucets with reduced water consumption
3	Washing machines with reduced water consumption
4	Raintanks for toilet and garden water use (3 kL)
5	Increment of the previous action with additional capacity (1.5 kL)
6	Raintanks for residential hot water (3 kL)
7	Small scale recycling for irrigation
8	Aquifer usage for irrigation of public open space
9	Dual reticulation system for recycling water

Table 3
Consequences of the alternatives in the sample problem.

Alt	P	N2	Climate	Savings	Cost	Water
	[T/yr]	[T/yr]	[-]	[MAUD]	[MAUD]	[%]
1	[0.9, 1.1]	[0.09, 0.11]	[0.0, 0.0]	[1.8, 2.2]	1.0	3.0
2	[1.1, 1.3]	[0.09, 0.11]	[0.0, 0.0]	[1.8, 2.2]	1.0	7.0
3	[1.3, 1.7]	[0.14, 0.17]	[0.5, 1.5]	[1.8, 2.2]	2.0	4.0
4	[0.0, 0.0]	[0.00, 0.00]	[-1.5, -0.5]	[1.8, 2.2]	10.	15.
5	[0.0, 0.0]	[0.00, 0.00]	[0.0, 0.0]	[0.9, 1.1]	8.0	10.
6	[0.0, 0.0]	[0.00, 0.00]	[0.5, 1.5]	[9.0, 11.]	11.	38.
7	[0.5, 0.6]	[0.00, 0.00]	[0.0, 0.0]	[32., 40.]	43.	15.
8	[0.0, 0.0]	[0.00, 0.00]	[0.0, 0.0]	[14., 18.]	23.	34.
9	[4.0, 4.8]	[0.40, 0.48]	[-2.5, -1.5]	[3.5, 4.5]	20.	46.

Table 4
Portfolio constraints in the test problem.

Constraint type	Description
Follow-up	Alternative 5 can only be included if Alternative 4 is included
Mutual exclusivity	Alternative 4 and Alternative 6 cannot be simultaneously present
Mutual exclusivity	Alternative 7 and Alternative 8 cannot be simultaneously present
Mutual exclusivity	Alternative 7 and Alternative 9 cannot be simultaneously present
Budget constraint	Overall cost should be less than 45 MAUD
Water target	Overall water reduction should be greater than 50%

of coefficient of variation (COV, Fig. 9). In this particular case, several non-dominated solutions (meaning that are optimal in the multi-objective optimisation sense) can be found in the set.

In addition, we include the proposed visualisation of the solutions in the non-dominated set (Fig. 10). Here we can observe the characteristics of the model structure, the results for the main and intermediate objectives, with an indication of its uncertainty. For example, it is possible to see that about half of the non-dominated solutions tend to produce low scores in the climate scenario, as indicated by the orange segments representing its value, being close to the start line in portfolios P2, P3, P5, P6 and P7 (remember that the value increases in counter-clockwise direction within each segment). Of the non-dominated portfolios, P8 clearly outperforms (by mean) the others as indicated by the darker color of the circle at the centre and position of the orange wedge within the circle (over 270° from origin).

We acknowledge the limitations in visualising many features simultaneously. Therefore, this visualisation proposes a way for the decision analyst of having an overall picture of the problem in terms of structure (weights, and hierarchy), results (values or utilities) and its uncertainty. We acknowledge that by no means this visualisation is aimed to obtain exact values (as wedges are scaled), and therefore, we see the benefits of presenting this figure alongside tables that describe the exact values of interest in the objectives, or by using interactive visualisations (such as mouseover displays or dynamic legends).

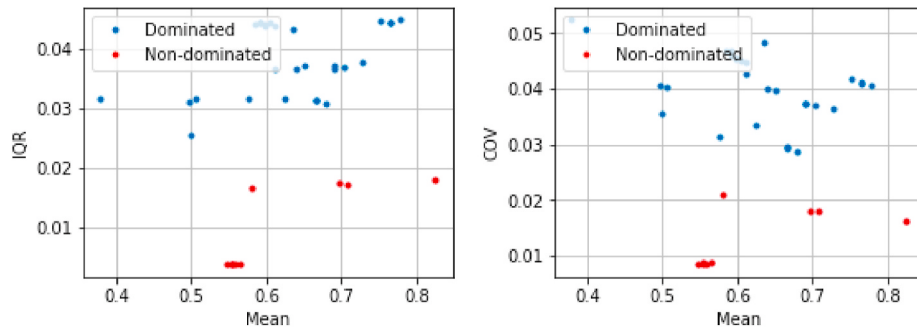


Fig. 9. Non-dominated (Pareto-optimal) portfolios based on expected value (mean), interquartile range (IQR), and coefficient of variation (COV).

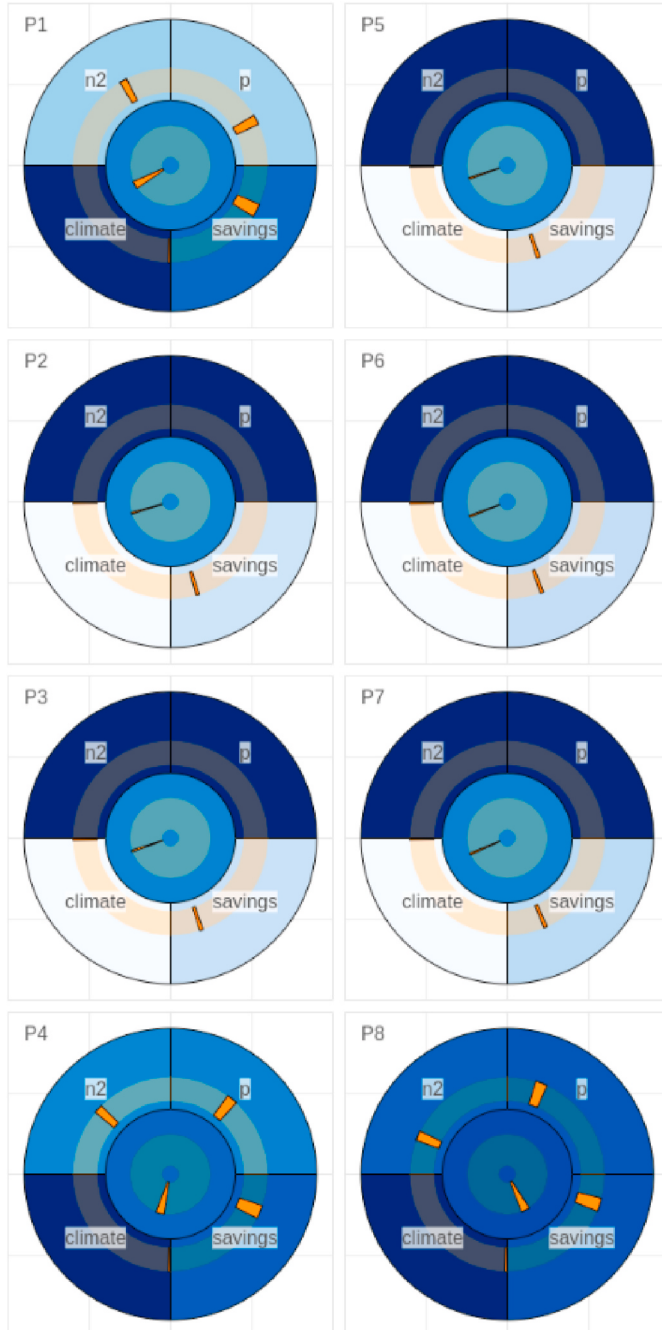


Fig. 10. Visualisation of the set of non-dominated (Pareto-optimal) solutions using the proposed visualisation alternative, for explanation see Fig. 5.

4.3. Limitations and potential for future extension

The current version of Decisi-o-rama includes only basic MAVT and MAUT models. While these are generally sufficient for exploratory research and applications, future users (including ourselves) can implement user-defined preference structures that current preference models cannot handle (Langhans et al., 2014; Langhans and Lienert, 2016; Reichert et al., 2019). More specifically, MAVT and MAUT assume rational decision makers, linear probability weighting and evaluation of alternatives on an absolute scale. These contradict observed decision-making behaviour which suggests, for example, presence of reference points, non-linear weighting of probabilities or ambiguous outcomes as well as different valuation of outcomes depending on whether these are framed as gains or losses, as reflected in Cumulative Prospect Theory (Stott, 2006). Beyond the modelling of preferences about outcomes and attached risks and uncertainties, incorporation of time-dependent preferences (Andreoni et al., 2015; Hermann and Musshoff, 2016) or social preferences (Chuang and Schechter, 2015; Galizzi and Navarro-Martinez, 2019) would also be necessary to advance the MCDA field both in terms of theory and practice.

Beyond preference modelling for individual actions, further developments will be needed to encompass more complicated portfolio preferences, such as multi-linear value functions for robust portfolio modelling (Liesiö, 2014) and portfolio-level measures of risk and regret (Liesiö et al., 2007; Vilkkumaa et al., 2014). Also, the example we presented did assume additivity (i.e. independence) between the value of actions in a portfolio. Further, more involved assessment models on attribute level will be needed to reflect dependencies in portfolio performance and contingencies in portfolio creation (Gustafsson and Salo, 2005; Liesiö et al., 2008).

Decisi-o-rama has two main limitations (that we can foresee) in the scope of its implementation that constraint its applicability in the view of more generic processes, or approaches, to the MAVT/MAUT problem. First, Decisi-o-rama does not support correlated or conditioned random variables in its generators, as instances of the generated random variables are created on the fly; however, this can be circumvented by passing complete vectors to represent the samples of the random variable instead of generators. Second, Decisi-o-rama only handles uncertainty in the form of probability distribution functions, and not in other encoding such as fuzzy sets or possibilistic distributions.

In the close future of Decisi-o-rama, we expect to add other features including sensitivity analysis and optimisation routines. In sensitivity analysis we see utility in developing local (one-at-a-time) and global sensitivity analysis for single portfolio, and model assessment respectively. In the optimisation routines, we expect to integrate heuristic optimisation methods that support exploring the potential portfolios (thus implying binary optimisation), as well as integer, continuous, and mixed-integer optimisation using linear and non-linear approaches.

The Decisi-o-rama library has been built keeping these possible developments in mind, ensuring flexibility to all types of extension and

efficiency in computation. We explicitly invite the community to build it further and to explore its usability in real world applications.

5. Conclusions

We have introduced Decisi-o-rama, an open-source Python library for MCDA on single-alternative and portfolio approaches, in the context of multi-attribute utility/value theory. We have demonstrated the capabilities of the tool to efficiently manage large multi-criteria decision analysis problems that include uncertain attributes and preference parameters using conventional hardware. Therefore, it can be used for exploratory modelling of decision alternatives with decision-makers on-the-fly. Also, due to its high computational efficiency, it can be used to support robust portfolio modelling applications, alternative optimisation algorithms, and sensitivity analysis, that may enrich the decision analysis process.

In addition, Decisi-o-rama offers a flexible implementation that supports user-defined preference models. In this respect, it is possible to define non-customary marginal value or utility functions as well as aggregation functions. Also, it is possible that the users specify uncertain parameters of the preference model through their user-defined distributions. These features ensure straightforward extensibility for different types of models and users.

Keeping in mind that accessibility and extensibility are paramount for future usability of Decisi-o-rama, it is offered to the community as open-source. This same principle was maintained through the development of the tool, reducing the number of third-party libraries for its execution and supporting the readability and maintainability of the code. Furthermore, the tool operates with regular Python objects, making its integration with other applications such as different attribute assessment (process) models and visualisation tools. With these decisions, we hope that this tool will be used in different settings and for different user groups, facilitating the adoption of MCDA methods by researchers and practitioners alike.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was conducted as part of the Cooperation Programme TISCA (Technology Innovation for Sewer Condition Assessment) with project number 15343, which is (partly) financed by NWO domain TTW (the domain applied and Engineering Sciences of the Netherlands Organisation for Scientific Research), the RIONED Foundation, STOWA (Foundation for Applied Water Research) and the Knowledge Program Urban Drainage (KPUD).

Furthermore, the authors would like to thank Julia Hartmann and Job van der Werf for their collaboration in reviewing this document, and the three anonymous reviewers whose constructive suggestions helped us in considerably improving the final version of this document.

References

- Andreoni, J., Kuhn, M.A., Sprenger, C., 2015. Measuring time preferences: a comparison of experimental methods. *J. Econ. Behav. Organ.* 116, 451–464. <https://doi.org/10.1016/j.jebo.2015.05.018>.
- Behzadian, M., Kazemzadeh, R., Albadvi, A., Aghdasi, M., 2010. PROMETHEE: a comprehensive literature review on methodologies and applications. *Eur. J. Oper. Res.* 200, 198–215. <https://doi.org/10.1016/j.ejor.2009.01.021>.
- Belton, V., Stewart, T.J., 2002. Multiple Criteria Decision Analysis. Springer US. <https://doi.org/10.1007/978-1-4615-1495-4>.
- Bigaret, S., Hodgett, R.E., Meyer, P., Mironova, T., Olteanu, A.L., 2017. Supporting the multi-criteria decision aiding process: R and the MCDA package. *EURO Journal on Decision Processes* 5, 169–194. <https://doi.org/10.1007/s40070-017-0064-1>.
- Blatter, C., Lemm, R., Thees, O., Lexer, M.J., Hanewinkel, M., 2017. Management of ecosystem services in mountain forests: review of indicators and value functions for model based multi-criteria decision analysis. *Ecol. Indic.* 79, 391–409. <https://doi.org/10.1016/j.ecolind.2017.04.025>.
- Bureš, V., Ponce, D., Čech, P., Mls, K., 2019. The effect of trial repetition and problem size on the consistency of decision making. *PLoS One* 14, e0216235. <https://doi.org/10.1371/journal.pone.0216235>.
- Calizaya, A., Meixner, O., Bengtsson, L., Berndtsson, R., 2010. Multi-criteria decision analysis (MCDA) for integrated water resources management (IWRM) in the lake poopo basin, Bolivia. *Water Resour. Manag.* 24, 2267–2289. <https://doi.org/10.1007/s11269-009-9551-x>.
- Chuang, Y., Schechter, L., 2015. Stability of experimental and survey measures of risk, time, and social preferences: a review and some new results. *J. Dev. Econ.* 117, 151–170. <https://doi.org/10.1016/j.jdevco.2015.07.008>.
- Cinelli, M., Coles, S.R., Kirwan, K., 2014. Analysis of the potentials of multi criteria decision analysis methods to conduct sustainability assessment. *Ecol. Indic.* 46, 138–148. <https://doi.org/10.1016/j.ecolind.2014.06.011>.
- Cinelli, M., Spada, M., Kim, W., Zhang, Y., Burgherr, P., 2020. MCDA index tool: an interactive software to develop indices and rankings. *Environment Systems and Decisions*. <https://doi.org/10.1007/s10669-020-09784-x>.
- Convertino, M., Baker, K., Vogel, J., Lu, C., Suedel, B., Linkov, I., 2013. Multi-criteria decision analysis to select metrics for design and monitoring of sustainable ecosystem restorations. *Ecol. Indic.* 26, 76–86. <https://doi.org/10.1016/j.ecolind.2012.10.005>.
- Durbach, I.N., Stewart, T.J., 2011. An experimental study of the effect of uncertainty representation on decision making. *Eur. J. Oper. Res.* 214, 380–392. <https://doi.org/10.1016/j.ejor.2011.04.021>.
- Dyer, J.S., 2005. MAUT — multiattribute utility theory. In: *International Series in Operations Research & Management Science*. Springer-Verlag, pp. 265–292. https://doi.org/10.1007/0-387-23081-5_7.
- Dyer, J.S., Sarin, R.K., 1979. Measurable multiattribute value functions. *Oper. Res.* 27, 810–822.
- Eisenführ, F., Weber, M., Langer, T., 2010. *Rational Decision Making*. Springer-Verlag GmbH.
- Esmail, B.A., Geneletti, D., 2018. Multi-criteria decision analysis for nature conservation: a review of 20 years of applications. *Methods in Ecology and Evolution* 9, 42–53. <https://doi.org/10.1111/2041-210X.12899>.
- Figueira, J., Ehrgott, M., Greco, S., 2006. *Multiple Criteria Decision Analysis: State of the Art Surveys*. Springer-Verlag GmbH.
- French, S., 2003. Modelling, making inferences and making decisions: the roles of sensitivity analysis. *Top* 11, 229–251. <https://doi.org/10.1007/bf02579043>.
- Galizzi, M.M., Navarro-Martinez, D., 2019. On the external validity of social preference games: a systematic lab-field study. *Manag. Sci.* 65, 976–1002. <https://doi.org/10.1287/mnsc.2017.2908.essential>.
- Gómez-Delgado, M., Tarantola, S., 2006. GLOBAL sensitivity analysis, GIS and multi-criteria evaluation for a sustainable planning of a hazardous waste disposal site in Spain. *Int. J. Geogr. Inf. Sci.* 20, 449–466. <https://doi.org/10.1080/13658810600607709>.
- Govindan, K., Jepsen, M.B., 2016. ELECTRE: a comprehensive literature review on methodologies and applications. *Eur. J. Oper. Res.* 250, 1–29. <https://doi.org/10.1016/j.ejor.2015.07.019>.
- Gustafsson, J., Salo, A., 2005. Contingent portfolio programming for the management of risky projects. *Oper. Res.* 53, 946–956. <https://doi.org/10.1287/opre.1050.0225>.
- Haag, F., Reichert, P., Maurer, M., Lienert, J., 2019. Integrating uncertainty of preferences and predictions in decision models: an application to regional wastewater planning. *J. Environ. Manag.* 252, 109652. <https://doi.org/10.1016/j.jenvman.2019.109652>.
- Hajkowicz, S., 2007. A comparison of multiple criteria analysis and unaided approaches to environmental decision making. *Environ. Sci. Pol.* 10, 177–184. <https://doi.org/10.1016/j.envsci.2006.09.003>.
- Hermann, D., Musshoff, O., 2016. Measuring time preferences: comparing methods and evaluating the magnitude effect. *Journal of Behavioral and Experimental Economics* 65, 16–26. <https://doi.org/10.1016/j.soec.2016.09.003.background>.
- Huang, I.B., Keisler, J., Linkov, I., 2011. Multi-criteria decision analysis in environmental sciences: ten years of applications and trends. *Sci. Total Environ.* 409, 3578–3594. <https://doi.org/10.1016/j.scitotenv.2011.06.022>.
- International Society on MCDM, 2020. Software related to mcdm. <https://www.mcdmso-ciety.org/content/software-related-mcdm>.
- Jones, D., Tamiz, M., 2016. A review of goal programming. In: *Multiple Criteria Decision Analysis*. Springer, New York, pp. 903–926. https://doi.org/10.1007/978-1-4939-3094-4_21.
- Joubert, A., Stewart, T.J., Eberhard, R., 2003. Evaluation of water supply augmentation and water demand management options for the city of cape town. *J. Multi-Criteria Decis. Anal.* 12, 17–25. <https://doi.org/10.1002/mcda.342>.
- Keeney, R.L., 1982. Feature article—decision analysis: an overview. *Oper. Res.* 30, 803–838. <https://doi.org/10.1287/opre.30.5.803>.
- Keeney, R.L., Howard, R., 2002. *Decisions with Multiple Objectives*. Cambridge University Press.
- Kiker, G.A., Bridges, T.S., Varghese, A., Seager, T.P., Linkov, I., 2005. Application of multicriteria decision analysis in environmental decision making. *Integrated Environ. Assess. Manag.* 1, 95. <https://doi.org/10.1897/ieam.2004a-015.1>.
- Kleinmuntz, J., 2007. *Resource Allocation Decisions*. Cambridge University Press, pp. 400–418.
- Kuemerlen, M., Reichert, P., Siber, R., Schuwirth, N., 2019. Ecological assessment of river networks: from reach to catchment scale. *Sci. Total Environ.* 650, 1613–1627. <https://doi.org/10.1016/j.scitotenv.2018.09.019>.

- Lahtinen, T.J., Hämäläinen, R.P., Liesjö, J., 2017. Portfolio decision analysis methods in environmental decision making. *Environ. Model. Software* 94, 73–86. <https://doi.org/10.1016/j.envsoft.2017.04.001>.
- Lam, S.K., Pitrou, A., Seibert, S., 2015. Numba: a llvm-based python jit compiler. In: *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC*. Association for Computing Machinery, New York, NY, USA. <https://doi.org/10.1145/2833157.2833162> doi:10.1145/2833157.2833162.
- Langhans, S.D., Lienert, J., 2016. Four common simplifications of multi-criteria decision analysis do not hold for river rehabilitation. *PLoS One* 11, e0150695. <https://doi.org/10.1371/journal.pone.0150695>.
- Langhans, S.D., Lienert, J., Schuwirth, N., Reichert, P., 2013. How to make river assessments comparable: a demonstration for hydromorphology. *Ecol. Indic.* 32, 264–275. <https://doi.org/10.1016/j.ecolind.2013.03.027>.
- Langhans, S.D., Reichert, P., Schuwirth, N., 2014. The method matters: a guide for indicator aggregation in ecological assessments. *Ecol. Indic.* 45, 494–507. <https://doi.org/10.1016/j.ecolind.2014.05.014>.
- Lattner, C., Adve, V., 2004. LLVM: a compilation framework for lifelong program analysis and transformation. In: *CGO, San Jose, CA, USA*, pp. 75–88.
- Lienert, J., Duygan, M., Zheng, J., 2016. Preference stability over time with multiple elicitation methods to support wastewater infrastructure decision-making. *Eur. J. Oper. Res.* 1, 746–760. <https://doi.org/10.1016/j.ejor.2016.03.010>.
- Lienert, J., Scholten, L., Egger, C., Maurer, M., 2014. Structured decision-making for sustainable water infrastructure planning and four future scenarios. *EURO Journal on Decision Processes* 3, 107–140. <https://doi.org/10.1007/s40070-014-0030-0>.
- Liesjö, J., 2014. Measurable multiattribute value functions for portfolio decision analysis. *Decis. Anal.* 11, 1–20. <https://doi.org/10.1287/deca.2013.0287> cited By :5 Export Date: 22 September 2016.
- Liesjö, J., Mild, P., Salo, A., 2007. Preference programming for robust portfolio modeling and project selection. *Eur. J. Oper. Res.* 181, 1488–1505. <https://doi.org/10.1016/j.ejor.2005.12.041>.
- Liesjö, J., Mild, P., Salo, A., 2008. Robust portfolio modeling with incomplete cost information and project interdependencies. *Eur. J. Oper. Res.* 190, 679–695. <https://doi.org/10.1016/j.ejor.2007.06.049>.
- Linkov, I., Moberg, E., 2017. *Multi-Criteria Decision Analysis: Environmental Applications and Case Studies*. Taylor & Francis Ltd.
- Marttunen, M., Lienert, J., Belton, V., 2017. Structuring problems for multi-criteria decision analysis in practice: a literature review of method combinations. *Eur. J. Oper. Res.* 263, 1–17. <https://doi.org/10.1016/j.ejor.2017.04.041>.
- Marttunen, M., Weber, C., Åberg, U., Lienert, J., 2019. Identifying relevant objectives in environmental management decisions: an application to a national monitoring program for river restoration. *Ecol. Indic.* 101, 851–866. <https://doi.org/10.1016/j.ecolind.2018.11.042>.
- Mendoza, G., Martins, H., 2006. Multi-criteria decision analysis in natural resource management: a critical review of methods and new modelling paradigms. *For. Ecol. Manag.* 230, 1–22. <https://doi.org/10.1016/j.foreco.2006.03.023>.
- Mild, P., Sahlo, A., 2009. Combining a multiattribute value function with an optimization model - an application to dynamic resource allocation for infrastructure maintenance. *Decis. Anal.* 6, 139–152. <https://doi.org/10.1287/deca.1090.0143>.
- Mustajoki, J., Marttunen, M., 2017. Comparison of multi-criteria decision analytical software for supporting environmental planning processes. *Environ. Model. Software* 93, 78–91. <https://doi.org/10.1016/j.envsoft.2017.02.026>.
- Nardo, M., Saisana, M., Saltelli, A., Tarantola, S., Hoffmann, A., Giovannini, E., 2008. *Handbook on Constructing Composite Indicators: Methodology and User Guide*. OECD, Paris.
- Oliphant, T., 2006. *Guide to NumPy*. Trelgol Publishing.
- Ortiz, G., Domínguez-Gómez, J.A., Aledo, A., Urgeghe, A.M., 2018. Participatory multi-criteria decision analysis for prioritizing impacts in environmental and social impact assessments. *Sustain. Sci. Pract. Pol.* 14, 6–21. <https://doi.org/10.1080/15487733.2018.1510237>.
- Reichert, P., Niederberger, K., Rey, P., Helg, U., Haertel-Borer, S., 2019. The need for unconventional value aggregation techniques: experiences from eliciting stakeholder preferences in environmental management. *EURO Journal on Decision Processes* 7, 197–219. <https://doi.org/10.1007/s40070-019-00101-9>.
- Reichert, P., Schuwirth, N., Langhans, S., 2013. Constructing, evaluating and visualizing value and utility functions for decision support. *Environ. Model. Software* 46, 283–291. <https://doi.org/10.1016/j.envsoft.2013.01.017>.
- Sa-nguanduan, N., Nititattananon, V., 2011. Strategic decision making for urban water reuse application: a case from Thailand. *Desalination* 268, 141–149. <https://doi.org/10.1016/j.desal.2010.10.010>.
- Salo, A., Keisler, J., Morton, A., 2011. *Portfolio Decision Analysis. Volume 162 of International Series in Operations Research & Management Science*. Springer, New York.
- Salo, A., Mild, P., Pentikäinen, T., 2006. Exploring causal relationships in an innovation program with robust portfolio modeling. *Technol. Forecast. Soc. Change* 73, 1028–1044. <https://doi.org/10.1016/j.techfore.2006.03.005>.
- Scholten, L., Maurer, M., Lienert, J., 2017. Comparing multi-criteria decision analysis and integrated assessment to support long-term water supply planning. *PLoS One* 12, e0176663. <https://doi.org/10.1371/journal.pone.0176663>.
- Scholten, L., Schuwirth, N., Reichert, P., Lienert, J., 2015. Tackling uncertainty in multi-criteria decision analysis – an application to water supply infrastructure planning. *Eur. J. Oper. Res.* 242, 243–260. <https://doi.org/10.1016/j.ejor.2014.09.044>.
- Singh, R.K., Murty, H., Gupta, S., Dikshit, A., 2012. An overview of sustainability assessment methodologies. *Ecol. Indic.* 15, 281–299. <https://doi.org/10.1016/j.ecolind.2011.01.007>.
- Stott, H., 2006. Cumulative prospect theory's functional menagerie. *J. Risk Uncertain.* 32, 101–130. <https://doi.org/10.1007/s11166-006-8289-6>.
- van der Walt, S., Colbert, S.C., Varoquaux, G., 2011. The numpy array: a structure for efficient numerical computation. *Comput. Sci. Eng.* 13, 22–30.
- Vilkumaa, E., Salo, A., Liesjö, J., 2014. Multicriteria portfolio modeling for the development of shared action agendas. *Group Decis. Negot.* 23, 49–70. <https://doi.org/10.1007/s10726-012-9328-0>.
- Virtanen, P., Gommers, R., Oliphant, T.E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S.J., Brett, M., Wilson, J., Jarrod Millman, K., Mayorov, N., Nelson, A.R.J., Jones, E., Kern, R., Larson, E., Carey, C., Polat, I., Feng, Y., Moore, E.W., van der Plas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E.A., Harris, C.R., Archibald, A. M., Ribeiro, A.H., Pedregosa, F., van Mulbregt, P., Contributors, S., et al., 2020. SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nat. Methods* 17, 261–272. <https://doi.org/10.1038/s41592-019-0686-2>.
- Voinov, A., Bousquet, F., 2010. Modelling with stakeholders. *Environ. Model. Software* 25, 1268–1281. <https://doi.org/10.1016/j.envsoft.2010.03.007> thematic Issue - Modelling with Stakeholders.
- Voinov, A., Kolagani, N., McCall, M.K., Glynn, P.D., Kragt, M.E., Ostermann, F.O., Pierce, S.A., Ramu, P., 2016. Modelling with stakeholders – next generation. *Environ. Model. Software* 77, 196–220. <https://doi.org/10.1016/j.envsoft.2015.11.016>.
- Von Neumann, J., Morgenstern, O., 2007. *Theory of Games and Economic Behavior*. Princeton University Press.
- Watt, D., 2006. *Programming Language Design Concepts*. Wiley. <https://books.google.nl/books?id=vogP3P2L4tgC>.
- Weistroffer, H.R., Li, Y., 2016. Multiple criteria decision analysis software. In: *Multiple Criteria Decision Analysis*. Springer, New York, pp. 1301–1341. https://doi.org/10.1007/978-1-4939-3094-4_29.
- Yee, K.P., van Rossum, G., 2001. PEP 234 - Iterators. Technical Report. Python Software Foundation.
- Zheng, J., Egger, C., Lienert, J., 2016. A scenario-based MCDA framework for wastewater infrastructure planning under uncertainty. *J. Environ. Manag.* 183, 895–908. <https://doi.org/10.1016/j.jenvman.2016.09.027>.