MCDA Calculator: A Streamlined Decision Support System for Multi-criteria Decision Analysis

He Huang(✉) and Peter Burgherr

Laboratory for Energy Systems Analysis, Paul Scherrer Institute, Forschungsstrasse 111, 5232 Villigen, PSI, Switzerland
{he.huang,peter.burgherr}@psi.ch

Abstract. The multi-criteria decision analysis (MCDA) landscape is fraught with complexity and challenges, particularly in diverse decision-making environments. Practitioners often face the challenging tasks of selecting appropriate MCDA methodologies, navigating complicated computational processes, and effectively synthesizing inputs from a variety of stakeholders. The existing landscape of MCDA tools, which are typically limited to specific methodologies, exacerbates these challenges, often resulting in fragmented workflows and steep learning curves. To overcome these hurdles, the MCDA Calculator (https://mcda-calculator.psi.ch) emerges as a novel decision support system (DSS), providing a unified and streamlined platform tailored to increase the efficiency and effectiveness of computational process for experienced practitioners in applying MCDA. The MCDA Calculator features a streamlined computational workflow that blends different MCDA methodologies into a cohesive unit. This approach ensures a consistent and intuitive user experience, effectively eliminating the need for complex, time-consuming configurations. The tool's design philosophy focuses on simplifying the MCDA calculation process. In this paper, we introduce our DSS and detail the workflow of the developed web-based tool. To illustrate the practical benefits and real-world applicability of the MCDA Calculator, the paper presents a numerical example which illustrates the tool's ability to streamline calculation processes, and produce insightful, actionable results.

Keywords: Multi-criteria decision analysis · Decision support system · Software development

1 Introduction

In the field of decision making, Multi-Criteria Decision Analysis (MCDA) methods have proliferated over several decades, forming a large and diverse "family" [15]. Each member of this family has distinct characteristics that make them applicable in different contexts. The diversity of MCDA methods provides practitioners with structured and context-specific approaches to evaluate complex decision problems. Additionally, the application of these methods in

S. P. Duarte et al. (Eds.): ICDSST 2024, LNBIP 506, pp. 31–45, 2024.
https://doi.org/10.1007/978-3-031-59376-5_3

real-world scenarios necessitates an intuitive and user-friendly interaction with the data, ideally through a Decision Support System (DSS). To address this, user interface(UI)-based software solutions have been developed that streamline the decision-making process [18]. Specialized software such as PriEsT for the Analytical Hierarchy Process (AHP) [26], Visual-PROMETHEE for the PROMETHEE method [21], and ValueDecisions for the Multi-Attribute Value Theory (MAVT) [16] have been developed, each tailored to the computational nuances of different MCDA methods. Some other software solutions are developed to guide and incorporate the input of multiple decision makers (DMs), experts, and stakeholders into the decision process. This requires an instructive framework that facilitates group decision-making and provides a comprehensive view of the collective results. Software solutions such as SOCRATES for the Social Multi-Criteria Evaluation (SMCE) framework [23] and MAMCA software [17] have been developed to focus on stakeholder interaction.

Despite advances in software tools designed to facilitate the application of MCDA methods, several gaps remain, particularly in their practical utility for practitioners. While most existing software provides detailed, instructive procedures to assist practitioners unfamiliar with these methods, these guidelines can be redundant and time-consuming for those already experienced in the field. The step-by-step instructions, while beneficial for beginners, can hinder the efficiency of experienced practitioners, causing unnecessary delays in their workflow. Another challenge is the need for versatility in applying different MCDA methods to different cases. Current software solutions are often dedicated to specific MCDA methodologies, forcing practitioners to switch between software platforms. This not only disrupts the continuity of their work, but also introduces additional learning curves as each software comes with its own unique interface and operating mechanics. This fragmentation of available tools can hinder the seamless integration of different MCDA approaches into a single, streamlined process. Although tools such as Diviz offer comprehensive solutions for the different MCDA methods, their effective utilization often necessitates a solid foundation in programming or an in-depth understanding of block building methodologies [4]. These prerequisites can create barriers for practitioners who may not possess such technical expertise, limiting the accessibility and wider adoption of these otherwise powerful MCDA tools.

To fill this gap, there is a need for a comprehensive tool that integrates different MCDA methodologies while providing intuitive and straightforward usability. Such a tool should minimize trivial settings and configurations, allowing practitioners, especially MCDA experts, to focus more on analysis and less on navigating the software. It should provide a platform where multiple MCDA methodologies can be seamlessly accessed and applied, with an easy-to-use interface that appeals to both novice and experienced practitioners. This consolidation of functionality would not only increase the efficiency of MCDA application, but also enrich the decision-making process, allowing for a more holistic and flexible approach to problem-solving in different contexts.

In this paper, our objective is to improve the methodological framework for MCDA computations, focusing specifically on integration within a visualized system. We present our newly developed DSS, the MCDA Calculator web tool, which is specifically designed to encapsulate different MCDA methods and seamlessly derive computations from them. We propose a streamlined computational structure for MCDA methods that allows different methods to adhere to a uniform process flow for generating results. This approach ensures that regardless of the MCDA method used, the process remains consistent and user-friendly, thereby simplifying the application of these methods and increasing the efficiency and effectiveness of the decision-making process. The paper is organized as follows: We begin with a literature review on existing MCDA methods and compare them with our software. Next, we detail the workflow and structure of the MCDA Calculator. Finally, we demonstrate the application of our tool through a numerical example, showcasing its practical utility.

2 Literature Review: Revisiting MCDA DSS from Practical Usage

In this section, much of the discourse on Multi-Criteria Decision Analysis (MCDA) revolves around its real-world application, particularly the challenges associated with its complex computational requirements. Given the intricate nature of MCDA calculations, there is an increasing reliance on computational support. For example, it can be facilitated by libraries in various programming languages. Notable examples include the Python-based libraries [7,30] and those in R [3]. Despite their comprehensiveness, these libraries present a significant hurdle for non-technical experts due to their lack of user-friendly interfaces and interactivity, making them less practical. Therefore, software as DSS play a critical role in bridging this gap. They help practitioners construct problem structures, implement MCDA methodologies, and visualize results in an accessible manner. Such systems offer increased convenience and flexibility in various contexts. For example, they allow practitioners to easily modify data and MCDA parameters or present results in stakeholder meetings and workshops. However, these DSSs are not without limitations. Each system has different features tailored to specific contexts. Some specialize in particular MCDA methods, while others emphasize interaction with practitioners, focus on post-hoc analysis, or are good at facilitating group decision making.

The variety of software underscores the importance of a comprehensive review to compare features and identify potential gaps. Such an analysis is critical to understanding their suitability in different decision-making scenarios. For this review, our attention is focused on MCDA DSSs with UI that have been developed within the last decade and remain accessible and operational today[1]. Given the large number of MCDA methods and software applications available, our focus will be on conducting a comparison of free (or partially free) MCDA tools for scoring and ranking. This comparative study aims to map the landscape of

[1] Test environment specifications: Operating System is Windows 10 and the processor is an Intel® Core™ i7-12800H.

available DSSs in MCDA, provide insights into their strengths and limitations, and identify areas where new developments could be most beneficial. Our comparative analysis focuses on the key features of different MCDA software, following the process stages proposed by Belton and Stewart [2]. These stages include problem structuring, model building, and challenging thinking. We examine and question the specific features offered by different software, focusing on how they facilitate each of these stages:

Phase 1: Problem structuring. The problem structuring phase is characterized by divergent thinking, where the focus is on mapping goals, values, and constraints while acknowledging the uncertainties and influences of external environmental factors. It is a stage where the range of stakeholders is brainstormed, along with the identification of primary alternatives and the establishment of appropriate criteria. The key features we examine are:

- $Q_{1,1}$: Does the DSS provide a heuristic approach to enable effective brainstorming for problem structuring?
- $Q_{1,2}$: Is the DSS designed to incorporate inputs from multiple actors, including stakeholders, decision-makers, or experts?

Phase 2: Model building. The model building phase marks a transition to a focused, convergent approach that synthesizes the rich insights from the problem structuring phase into defined, actionable elements of the decision process. This phase focuses on detailing alternatives, establishing criteria, and capturing associated values, preferences, and performance. We evaluate the following key features of the DSS:

- $Q_{2,1}$: Does the DSS provide a method for importing structured data to streamline the process instead of manual information entry?
- $Q_{2,2}$: Does the DSS provide weight elicitation methods (rather than directly entering the criteria weights)?
- $Q_{2,3}$: What MCDA method(s) does the DSS offer?

Phase 3: Challenging Thinking This phase brings the model results into discussion and uses critical analysis to explore the robustness of the constructed models to different scenarios and assumptions. For example, sensitivity analysis can be used to understand the impact of changes in criteria weights and alternative performance, and to assess the stability of the decision outcome. This phase ensures the robustness and reliability of the decision process. We evaluate the following key features of the DSS:

- $Q_{3,1}$: Does the DSS offer visualization tools for results to aid in discussion and interpretation?
- $Q_{3,2}$: Is there a feature within the DSS to export the results of the MCDA for further use and analysis?
- $Q_{3,3}$: Does the DSS include a module for conducting sensitivity analysis to assess the robustness of the decision outcomes?

In addition, we ask some general questions about the nature of software:

- $Q_{0,1}$: Does the platform operate as a web-based tool or is it configured for desktop installation?
- $Q_{0,2}$: Is the software open source?

The detailed comparative analysis of the MCDA DSSs is presented in Table 1.

Table 1. MCDA DSS comparison

DSS	$Q_{0,1}$	$Q_{0,2}$	$Q_{1,1}$	$Q_{1,2}$
Entscheidungsnavi [24]	Web-based	✓	✓	✗
FITradeoff [14]	Web-based	✗	✗	✗
MAMCA [17]	Web-based	✗	✗	✓
MCDA Index Tool [9]	Web-based	✗	✗	✗
PriEsT [25]	Desktop-based	✓	✗	✗
SOCRATES [23]	Web-based	✗	✗	✓
ValueDecisions [16]	Web-based	✓	✗	✓
Visual PROMETHEE [21]	Desktop-based	✗	✗	✗

DSS	$Q_{2,1}$	$Q_{2,2}$	$Q_{2,3}$
Entscheidungsnavi	JSON	✓	MAUT [12]
FITradeoff	Excel	✓	FITradeoff [13]
MAMCA	Excel	✓	AHP [28], SMART [29]
MCDA Index Tool	CSV	✓	SAW [19]
PriEsT	Special format	✓	AHP [28]
SOCRATES	JSON	✗	NAIADE [22]
ValueDecisions	Excel	✗	MAVT [5]
Visual PROMETHEE	CSV, TXT	✗	PROMETHEE [6]

DSS	$Q_{3,1}$	$Q_{3,2}$	$Q_{3,3}$
Entscheidungsnavi	✓	JSON	✓
FITradeoff	✓	Excel	✓
MAMCA	✓	Excel	✓
MCDA Index Tool	✓	CSV	✓
PriEsT	✓	Special format	✓
SOCRATES	✓	JSON	✓
ValueDecisions	✓	✗	✓
Visual PROMETHEE	✓	CSV, etc.	✓

In general, there is a trend in MCDA DSSs to move from traditional desktop-based applications to web-based tools. This shift addresses the various environmental support requirements, such as specific Java libraries, that desktop applications require. Web tools offer ease of use; a simple browser is all that is needed to launch the DSS, enabling cross-platform compatibility across PCs, mobile phones and tablets. Import and export functionality is considered essential, as evidenced by its ubiquity in all DSSs surveyed. However, the preferred formats for these functions vary, with easy-to-use options such as CSV and Excel being more common, as opposed to JSON or proprietary formats that can pose usability challenges.

In addition, it is observed that each DSS tends to specialize in different MCDA methodologies, resulting in different data entry methods and value elicitation processes. This specialization highlights the need for a tailored approach to handle the nuances of different MCDA techniques. Finally, the provision of sensitivity analysis is a notable feature, whose importance is underscored by its consistent inclusion in all DSSs. The prevalence of this feature indicates its crucial role in assessing the robustness of decision outcomes. Additionally, several DSSs also integrate group decision-making, which makes collaborative decision-making possible.

Our analysis shows that modern MCDA DSSs are equipped with comprehensive functionalities that enable a complete decision process within the software. In particular, Entscheidungsnavi provides a heuristic approach to problem structuring. However, we observe that these systems often have a steep learning curve. Typically, DSSs are limited to one or two MCDA methods, a limitation resulting from the inherent diversity of MCDA methods. As a result, practitioners may have to switch between different DSS platforms to use different MCDA methods. In addition, as shown in Fig. 1, the functional design of these DSSs varies, resulting in different workflows. Some systems adopt a linear process flow that requires step-by-step input from practitioners. While systematic, this approach can be time-consuming. Others offer a comprehensive set of functions across different sections of the interface, which, while thorough, can be confusing due to its complexity.

While the current landscape of DSSs with comprehensive MCDA capabilities is impressive, it often exceeds the needs of practitioners seeking speed and flexibility. There is a notable research gap in addressing scenarios where practitioners require fast, straightforward computations using different MCDA methods across different case studies. Existing systems, with their extensive feature sets and structured workflows, are designed more for in-depth analysis, which, while thorough, can be cumbersome for practitioners who need to quickly switch between methods and case studies. This gap highlights the need for a more agile and adaptable DSS that prioritizes efficiency and ease of use without compromising the breadth of MCDA methodologies.

3 Proposal of a Streamlined MCDA DSS: MCDA Calculator

In response to the research gap we identified, we present our proposed DSS: the MCDA Calculator. The goal is to provide a systematic and streamlined framework to assist practitioners in developing computational structure and efficiently

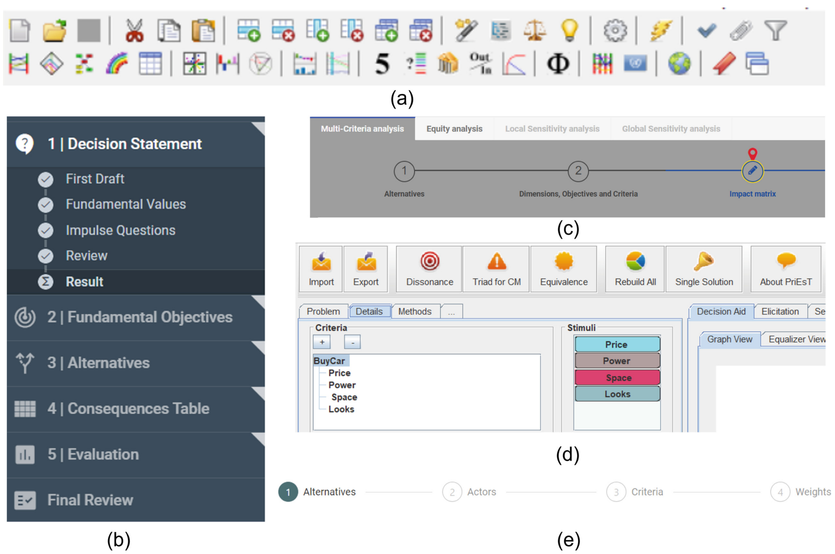

Fig. 1. Screenshots from MCDA DSSs: (a) Visual PROMETHEE; (b) Entscheidungsnavi; (c) SOCRATES; (d) PriEsT; (e) MAMCA.

computing results using a range of MCDA methodologies according to a set of defined requirements: Through our investigation, we discovered that a certain category of MCDA methods could be effectively incorporated into a unified computational model. The defining characteristics of these MCDA methods include:

- The primary focus is on ranking problems, where the goal is to obtain a complete ordinal ranking of alternatives;
- The sets of alternatives and criteria are predetermined and fixed, ensuring a stable and consistent data structure;
- Scoring functions are applied to integrate multiple criteria, culminating in a comprehensive final ranking.

Several prominent MCDA methods, such as PROMETHEE [6], TOPSIS [1], and MAVT [5], satisfy these criteria. Thus, in our DSS, we propose a high-level computational model, denoted $\mathcal{F}$, for these MCDA methods, taking into account various parameters:

$$\mathcal{F}(\mathcal{A}, \mathcal{M}, \mathcal{P}, \mathcal{G}), \tag{1}$$

where $\mathcal{A}$ denotes the matrix of collected alternative data over different criteria. $\mathcal{M}$ is a 1-tuple that specifically indicates the chosen MCDA method. $\mathcal{P}$ is the matrix that encapsulates parameters for the criteria, such as polarity, weights of the criteria, and method-specific parameters like those used in PROMETHEE to construct preference functions. Finally, $\mathcal{G}$ symbolizes the global parameters, represented as a tuple. This can range from a 0-tuple to n-tuples, depending on the specific MCDA method used. For example, in the context of VIKOR, the global parameter v is given to define the decision strategy [20], resulting in a 1-tuple for global parameters.

To clarify, the matrices $\mathcal{A}$ and $\mathcal{P}$ are described in a manner consistent with computer science principles, accommodating a range of data types and scales, including both numeric and non-numeric (e.g., strings) elements. Their representation as matrices serves primarily to structure the data in an organized and accessible format. The operations within the computational model extend beyond conventional mathematical matrix operations to include specialized transformations and mappings relevant to each MCDA method.

The primary goal of the computational model is to provide aggregate scores for the alternatives under consideration. However, in certain MCDA methods, additional information may prove valuable. For example, in the PROMETHEE method, we encounter metrics such as negative flows, positive flows, and net flows, while TOPSIS provides distances relative to the best and worst conditions. To effectively capture and use this additional information, we propose to structure it as matrices, denoted by $\mathcal{R}$. This approach not only increases the comprehensiveness of our analysis, but also facilitates integration with the programming languages used to develop our DSS.

With the computational model in place, we now present our MCDA calculator flowchart. As shown in Fig. 2, our proposed MCDA calculator is defined by a linear workflow, but includes decision points where the practitioner must ensure that the correct data or parameters have been entered before proceeding to the next step. This ensures the accuracy and completeness of the information entered at each stage of the MCDA process.

First, practitioners import the performance data of alternatives into the data matrix $\mathcal{A}$. The DSS expects an $x \times y$ matrix, i.e., $\mathcal{A}_{x \times y}$, for a decision problem consisting of x alternatives and y criteria. If the data matrix is not properly structured, the system prompts the practitioner to adjust the data. Once the data is properly formatted, the next step is to select an appropriate MCDA method, represented by $\mathcal{M}$. After selecting the MCDA method, the practitioner must enter the necessary criteria parameters in the matrix $\mathcal{P}$ and global parameters in $\mathcal{G}$, if applicable. For an MCDA method that requires z criterion parameters, the DSS expects a $y \times z$ matrix, i.e., $\mathcal{M}_{y \times z}$. Only when all parameters are correctly filled, the system proceeds to calculate the MCDA scores by applying the model $\mathcal{F}(\mathcal{A}, \mathcal{M}, \mathcal{P}, \mathcal{G})$, and finally obtains the MCDA result matrix $\mathcal{R}$. Should practitioners or DMs find the results unsatisfactory, or if they wish to validate the performance of alternatives using different MCDA methods, they have the option to recalculate. Importantly, this can be done without re-importing all the data, streamlining the process for further analysis.

4 Development and Demonstration of the Web-Based DSS

Based on the structure outlined in the flowchart, we developed the web-based DSS, the MCDA Calculator. This application was built using Dash, a flexible and lightweight Python framework designed for building web applications [10]. The MCDA Calculator is hosted at https://mcda-calculator.psi.ch and operates as a one-page application. This design allows practitioners to experience a cohesive workflow on a single page, encompassing every step from the initial data import to the final result (see Fig. 4). Currently, the MCDA calculator includes methods like PROMETHEE, TOPSIS, VIKOR, and SMART.

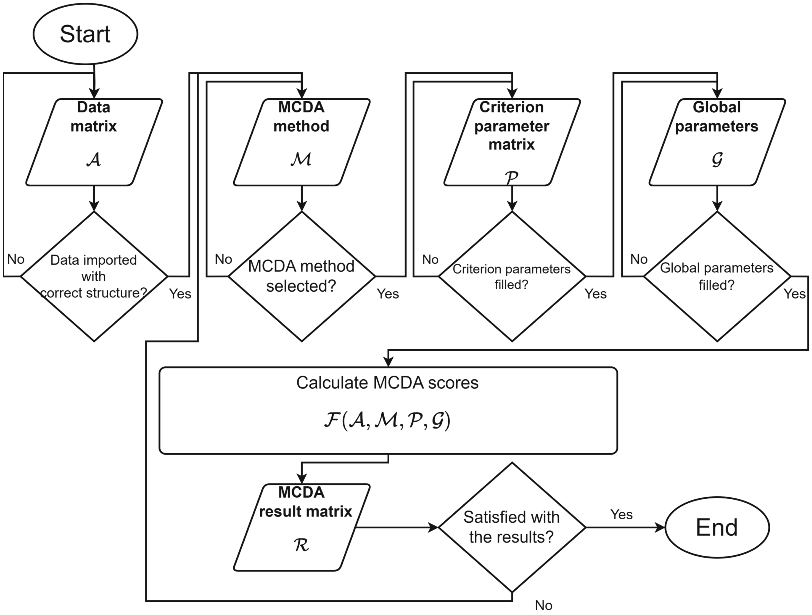

Fig. 2. MCDA Calculator flowchart

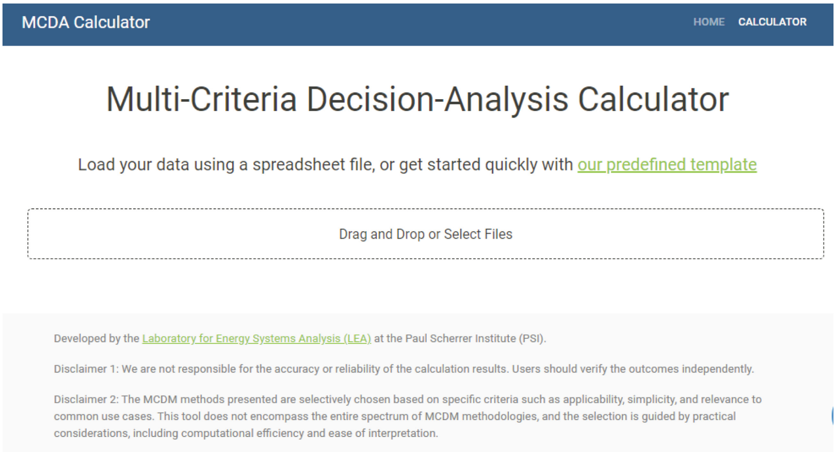

Fig. 3. Screenshot of the MCDA Calculator DSS

Given the abstract nature of the computational model, we will demonstrate the workflow of the DSS with a simplified analysis, which was originally designed as a practical exercise for master's students in energy system analysis at ETH Zurich. The case is adapted from a published work of Siskos and Burgherr [27], which analyzes the resilience of European countries' electricity supply systems. It aims to deepen the understanding of how complex and multidimensional concepts like energy system resilience can be assessed using a MCDA methodology. The exercise focuses on the evaluation of three major resilience dimensions: Resist, Restabilise, and Recover. As a simplified case study, the task involves evaluating and ranking Switzerland and its neighboring European countries based on their performance across 6 criteria (indicators) that influence their electricity supply resilience. The criteria are:

1. c_1 System Average Interruption Duration Index - SAIDI (Resist dimension): The SAIDI is a measure of the total duration of electricity supply interruptions per customer per year.
2. c_2 Political stability and absence of violence/terrorism (Resist dimension): A composite indicator that measures to what extent the political system is stable and not hindered by acts of violence and terrorism.
3. c_3 Electricity mix diversity (Restabilise dimension): This criterion measures the diversification of the electricity mix of each country, to different electricity generation technologies.
4. c_4 Electricity import dependence (Restabilise dimension): This criterion defines the ratio between electricity consumption and production in each European country.
5. c_5 Annual GDP growth (Recover dimension): A country exhibiting an expansion to its Gross Domestic Product (GDP) is expected to foster long-term investments and economic growth.
6. c_6 Government effectiveness (Recover dimension): Government effectiveness represents the quality of public services and the readiness of policy formulation and implementation.

The collected data for the alternative are depicted in Table 2. It should be noted that the majority of the indicators presented are composite indices. Consequently, specific units are not assigned to these indicators in the table.

Table 2. Data for evaluation

	c_1	c_2	c_3	c_4	c_5	c_6
Switzerland	0.20	1.34	0.43	0.98	2.05	2.04
Germany	0.30	0.60	0.77	0.92	1.93	1.62
France	0.40	0.11	0.47	0.89	1.52	1.48
Italy	1.30	0.31	0.73	1.15	0.90	0.41
Austria	0.60	0.92	0.56	1.14	1.89	1.45

To compute the resilience scores of different countries (alternatives) based on specific criteria, we will use two intuitive MCDA methods available in our DSS and demonstrate their efficient workflow. For this demonstration, we have selected methods from the Simple Multi-Attribute Rating Technique (SMART) family and the Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS). Readers interested in a comprehensive explanation of these methods can refer to the existing literature [1,11].

The first step within the DSS is to upload the data matrix $\mathcal{A}$ as described in Table 2. Practitioners have the convenience of importing data via a spreadsheet compatible with Excel or CSV formats. After importing the data, a method selection prompt appears, allowing the practitioner to select an MCDA method. When SMART is selected, the system dynamically generates parameter tables tailored for user-friendly input, allowing practitioners to populate the parameter matrix $\mathcal{P}$. For each criterion, practitioners must specify the corresponding polarity, minimum plausible parameter $q_{\min}$, maximum plausible parameter $q_{\max}$, and weight, as described in Table 3. As there is no global parameter in SMART, after completing this step, the DSS automatically skips this step, and practitioners can proceed to calculate the results by clicking the "Calculate the Result" button. When practitioners choose the TOPSIS method, the required parameters for $\mathcal{P}$ include the polarities of the criteria and their corresponding weights (see Table 3). There is no need for practitioners to start from scratch; they can seamlessly transition to calculating the TOPSIS results after calculating the SMART results, ensuring a smooth and efficient analysis process.

Table 3. Criterion parameters in SMART and TOPSIS method

	SMART parameters				TOPSIS parameters	
Criterion	Polarity	$q_{\min}$	$q_{\max}$	Weight	Polarity	Weight
c_1	Negative	0,2	1,3	0,08	Negative	0,08
c_2	Positive	0,11	1,34	0,2	Positive	0,2
c_3	Positive	0,43	0,77	0,12	Positive	0,12
c_4	Negative	0,89	1,15	0,2	Negative	0,2
c_5	Positive	0,9	2,05	0,16	Positive	0,16
c_6	Positive	0,41	2,04	0,24	Positive	0,24

The DSS will then calculate the SMART and TOPSIS results. The resulting matrix shows different results after computation. Within the SMART method, we document both the aggregated scores for a comprehensive overview and the normalized scores across all criteria to facilitate detailed analysis. In the TOPSIS methodology, we document the aggregated scores, and the distances of the alternatives to the ideal alternative as well as to the negative ideal alternative (nadir). In addition, we record the normalized scores for each criterion. Consequently, the matrix $\mathcal{R}$ is constructed to represent this data, as shown in Table 4. Practitioners have the option to export the result matrix, along with any other matrices, as Excel files for further in-depth analysis.

Table 4. Comparative Results $\mathcal{R}$ using SMART and TOPSIS Methods

(a) Results of the SMART Method

Alternative	Score	c_1	c_2	c_3	c_4	c_5	c_6
Switzerland	81.08	8.00	20.00	0.00	13.08	16.00	24.00
Germany	77.08	7.27	7.97	12.00	17.69	14.33	17.82
France	52.34	6.55	0.00	1.41	20.00	8.63	15.75
Italy	13.84	0.00	3.25	10.59	0.00	0.00	0.00
Austria	52.71	5.09	13.17	4.59	0.77	13.77	15.31

(b) Results of the TOPSIS Method

Alternative	Score	Distance to Ideal	Distance to Nadir	c_1	c_2	c_3	c_4	c_5	c_6
Switzerland	0.86	0.03	0.20	0.13	0.76	0.32	0.43	0.54	0.61
Germany	0.59	0.09	0.13	0.20	0.34	0.57	0.40	0.50	0.48
France	0.39	0.15	0.10	0.26	0.06	0.35	0.39	0.40	0.44
Italy	0.16	0.18	0.03	0.85	0.18	0.54	0.50	0.24	0.12
Austria	0.64	0.07	0.13	0.39	0.52	0.41	0.50	0.49	0.43

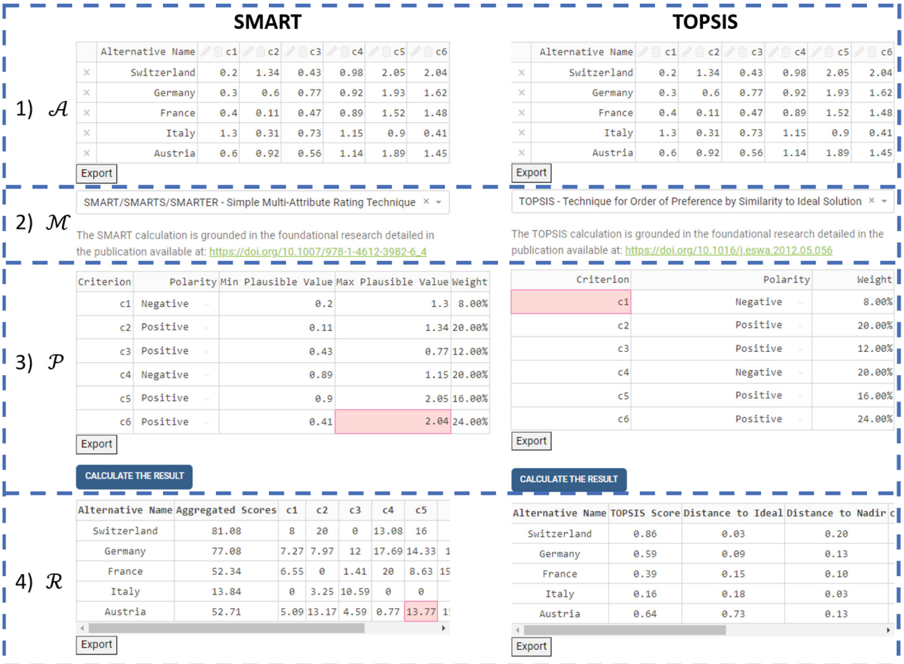

Fig. 4. DSS Workflow with SMART and TOPSIS

We can elaborate how the calculation of SMART and TOPSIS is done in our computational model $\mathcal{F}$ in the web-based DSS, as illustrated in Fig. 3. Practitioners can leverage the MCDA Calculator's linear and reversible workflow to easily adjust parameters or compare results across MCDA methods. By streamlining the calculation process and providing essential functionality in an accessible format, the MCDA Calculator stands out as a practical, time-saving tool in the field of MCDA, filling a critical gap in the current landscape of DSS.

5 Conclusion and Outlook

In this study, we have introduced the MCDA Calculator, a streamlined, web-based DSS specifically designed for facilitating MCDA calculations. This tool is optimized to support the MCDA process by simplifying the workflow and focusing primarily on delivering calculation results and associated parameters. Practitioners are guided through four straightforward steps, from data import to the exportation of MCDA results, making it particularly beneficial for scenarios requiring swift MCDA computations across different methods without the need to switch between various software platforms.

However, it is important to acknowledge that the MCDA Calculator is still in its developmental stages, presenting significant opportunities for enhancement. A key direction for future development involves expanding the range of MCDA methods incorporated within the system. As the MCDA-MSS suggests, there are up to 65 methods (including original MCDA methods and their variants) compatible with our DSS's framework [8]. This indicates a promising direction for extending the calculator's functionality.

While the MCDA Calculator features computational capabilities, the MCDA process encompasses additional aspects such as weighting and sensitivity analysis. Herein lies the potential to evolve the MCDA calculator into a comprehensive, modular DSS. This system would retain the streamlined computational workflow of the MCDA Calculator while integrating it with other modules to handle different aspects of the MCDA process. Such an advancement would not only maintain the efficiency of the MCDA Calculator, but also expand its scope to cover a wider range of MCDA functionalities, providing a holistic and versatile tool for multi-criteria decision analysis.

Acknowledgments. The research published in this article was carried out with the support of the Swiss Federal Office of Energy (SFOE) as part of the SWEET project SURE. The authors bear sole responsibility for the conclusions and the results presented in this publication.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Behzadian, M., Otaghsara, S.K., Yazdani, M., Ignatius, J.: A state-of the-art survey of topsis applications. Expert Syst. Appl. **39**(17), 13051–13069 (2012)
2. Belton, V., Stewart, T.: Multiple criteria decision analysis: an integrated approach. Springer Science & Business Media (2002)

3. Bigaret, S., Hodgett, R.E., Meyer, P., Mironova, T., Olteanu, A.L.: Supporting the multi-criteria decision aiding process: R and the mcda package. EURO J. Decision Proc. **5**(1–4), 169–194 (2017)
4. Bigaret, S., Meyer, P.: Diviz: an mcda workflow design, execution and sharing tool. Newsletter of the EURO Working Group Multicriteria Aid Decisions **3**(21), 10–13 (2010)
5. Bottero, M., Ferretti, V., Mondini, G.: Constructing multi-attribute value functions for sustainability assessment of urban projects. In: Murgante, B., et al. (eds.) ICCSA 2014. LNCS, vol. 8581, pp. 51–64. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-09150-1_5
6. Brans, J.P., De Smet, Y.: Promethee methods. Multiple criteria decision analysis: state of the art surveys, pp. 187–219 (2016)
7. Chacon-Hurtado, J.C., Scholten, L.: Decisi-o-rama: an open-source python library for multi-attribute value/utility decision analysis. Environm. Model. Softw. **135**, 104890 (2021)
8. Cinelli, M., Kadziński, M., Miebs, G., Gonzalez, M., Słowiński, R.: Recommending multiple criteria decision analysis methods with a new taxonomy-based decision support system. Eur. J. Oper. Res. **302**(2), 633–651 (2022)
9. Cinelli, M., Spada, M., Kim, W., Zhang, Y., Burgherr, P.: Mcda index tool: an interactive software to develop indices and rankings. Environ. Syst. Decis. **41**(1), 82–109 (2021)
10. Dabbas, E.: Interactive Dashboards and Data Apps with Plotly and Dash: Harness the power of a fully fledged frontend web framework in Python–no JavaScript required. Packt Publishing Ltd.(2021)
11. Edwards, W.: How to use multiattribute utility measurement for social decision-making. IEEE Trans. Syst. Man Cybern. **7**(5), 326–340 (1977)
12. Farquhar, P.H.: State of the art-utility assessment methods. Manage. Sci. **30**(11), 1283–1300 (1984)
13. Fossile, D.K., Frej, E.A., da Costa, S.E.G., de Lima, E.P., de Almeida, A.T.: Selecting the most viable renewable energy source for brazilian ports using the fitradeoff method. J. Clean. Prod. **260**, 121107 (2020)
14. Frej, E.A., de Almeida, A.T., Costa, A.P.C.S.: Using data visualization for ranking alternatives with partial information and interactive tradeoff elicitation. Oper. Res. Int. J. **19**, 909–931 (2019)
15. Greco, S., Figueira, J., Ehrgott, M.: Multiple criteria decision analysis, vol. 37. Springer (2016)
16. Haag, F., Aubert, A.H., Lienert, J.: Valuedecisions, a web app to support decisions with conflicting objectives, multiple stakeholders, and uncertainty. Environ. Model. Softw. **150**, 105361 (2022)
17. Huang, H., Lebeau, P., Macharis, C.: The multi-actor multi-criteria analysis (mamca): new software and new visualizations. In: International Conference on Decision Support System Technology, pp. 43–56. Springer (2020)
18. Ishizaka, A., Nemery, P.: Multi-criteria decision analysis: methods and software. John Wiley & Sons (2013)
19. Kaliszewski, I., Podkopaev, D.: Simple additive weighting-a metamodel for multiple criteria decision analysis methods. Expert Syst. Appl. **54**, 155–161 (2016)
20. Mardani, A., Zavadskas, E.K., Govindan, K., Amat Senin, A., Jusoh, A.: Vikor technique: a systematic review of the state of the art literature on methodologies and applications. Sustainability **8**(1), 37 (2016)

21. Mareschal, B., De Smet, Y.: Visual promethee: Developments of the promethee & gaia multicriteria decision aid methods. In: 2009 IEEE International Conference on Industrial Engineering and Engineering Management, pp. 1646–1649. IEEE (2009)
22. Munda, G.: Multicriteria evaluation in a fuzzy environment: The naiade method. Multicriteria Evaluation in a Fuzzy Environment: Theory and Applications in Ecological Economics, pp. 131–148 (1995)
23. Munda, G.: Qualitative reasoning or quantitative aggregation rules for impact assessment of policy options? a multiple criteria framework. Qual. Quant. **56**(5), 3259–3277 (2022)
24. von Nitzsch, R., Tönsfeuerborn, M., Siebert, J.U.: Decision skill training with the Entscheidungsnavi. In: de Almeida, A.T., Morais, D.C. (eds.) INSID 2020. LNBIP, vol. 405, pp. 15–30. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-64399-7_2
25. Siraj, S., Leonelli, R.C., Keane, J.A., Mikhailov, L.: Priest: a tool to estimate priorities from inconsistent judgments. In: 2013 IEEE International Conference on Systems, Man, and Cybernetics, pp. 44–49. IEEE (2013)
26. Siraj, S., Mikhailov, L., Keane, J.A.: Priest: an interactive decision support tool to estimate priorities from pairwise comparison judgments. Int. Trans. Oper. Res. **22**(2), 217–235 (2015)
27. Siskos, E., Burgherr, P.: Multicriteria decision support for the evaluation of electricity supply resilience: exploration of interacting criteria. Eur. J. Oper. Res. **298**(2), 611–626 (2022)
28. Vaidya, O.S., Kumar, S.: Analytic hierarchy process: an overview of applications. Eur. J. Oper. Res. **169**(1), 1–29 (2006)
29. Valiris, G., Chytas, P., Glykas, M.: Making decisions using the balanced scorecard and the simple multi-attribute rating technique. Perform. Meas. Metrics **6**(3), 159–171 (2005)
30. Wątróbski, J., Bączkiewicz, A., Sałabun, W.: pyrepo-mcda-reference objects based mcda software package. SoftwareX **19**, 101107 (2022)