Enhancing decision analysis with a large language model: pyDecision a comprehensive library of MCDA methods in Python

Journal of Modelling in Management

Valdecy Pereira, Marcio Pereira Basilio and Carlos Henrique Tarjano Santos
Department of Production Engineering, Federal Fluminense University, Niteroi, Brazil

Received 21 April 2024
Revised 12 January 2025
Accepted 10 June 2025

Abstract

Purpose – Multicriteria decision analysis (MCDA) has become increasingly essential for decision-making in complex environments. In response to this need, the pyDecision library, implemented in Python and available at https://bit.ly/3tLFGtH, has been developed to provide a comprehensive and accessible collection of MCDA methods. The purpose of this paper is to demonstrate the capabilities of pyDecision and illustrate how it can be effectively applied to solve multicriteria decision problems.

Design/methodology/approach – The pyDecision offers 70 MCDA methods, including analytic hierarchy process, technique for order of preference by similarity to ideal solution, the preference ranking organization method for enrichment evaluation and ELimination Et Choix Traduisant laRÉalité families. Beyond offering a vast range of techniques, the library provides visualization tools for more intuitive results interpretation. In addition to these features, pyDecision has integrated ChatGPT, an advanced large language model, where decision-makers can use ChatGPT to discuss and compare the outcomes of different methods, providing a more interactive and intuitive understanding of the solutions.

Findings – Large language models are undeniably potent but can sometimes be a double-edged sword. Its answers may be misleading without rigorous verification of its outputs, especially for researchers lacking deep domain expertise. It is imperative to approach its insights with a discerning eye and a solid foundation in the relevant field.

Originality/value – With the integration of MCDA methods and ChatGPT, pyDecision is a significant contribution to the scientific community, as it is an invaluable resource for researchers, practitioners and decision-makers navigating complex decision-making problems and seeking the most appropriate solutions based on MCDA methods.

Keywords Multicriteria decision analysis, Python, Decision-making

Paper type Research paper

1. Introduction

Multicriteria decision analysis (MCDA) is a decision-making methodology widely used in many fields, such as finance, engineering, environmental science and public policy. For instance, Pendokhare *et al.* (2024) demonstrated the successful application of MCDA techniques to optimize electrical discharge machining processes, while Kalita *et al.* (2022) used an MCDA-based approach to address parametric optimization in abrasive jet machining. Similarly, Trivedi *et al.* (2024) used MCDA to effectively prioritize road safety improvement measures, showcasing the versatility of these techniques across diverse domains. It is instrumental when decision-makers must make complex decisions based on multiple criteria, so MCDA methods can help decision-makers identify the best alternative

Journal of Modelling in Management

1746-5664
DOI 10.1108/JM2-04-2024-0118

among feasible options. However, the increasing complexity of decision-making problems across various domains has increased the demand for robust, easy-to-use tools (Taherdoost and Madanchian, 2023; Pereira *et al.*, 2017).

Figure 1 illustrates a generic MCDA framework that guides decision-making through structured steps. The process begins with "Defining Objectives and Criteria," where decision-makers outline the purpose of the decision and establish criteria for evaluating alternatives. Objectives serve as a roadmap for what the decision aims to achieve, while criteria provide measurable or qualitative aspects to assess options, aligning with the decision's purpose and stakeholder priorities. The next step, "Identifying Alternatives," involves listing all potential options for evaluation. These alternatives represent the competing choices for achieving the objectives, and a comprehensive list ensures no viable option is overlooked. The third step, "Collecting Data," focuses on gathering relevant information about the alternatives and criteria, including quantitative data (e.g. costs, benefits and performance scores) and qualitative inputs (e.g. expert opinions and stakeholder feedback). Accurate and reliable data are critical to ensure a fair evaluation. In the fourth step, "Choosing the MCDA Method," an appropriate decision-making technique is selected based on the problem's context, complexity and preferences. Popular methods include Analytic Hierarchy Process (AHP), Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) and VIseKriterijumska Optimizacija I Kompromisno Resenje (VIKOR), each influencing how criteria are weighted, alternatives are evaluated and results are derived. The fifth step, "Evaluating Alternatives," applies the chosen MCDA method to assess options against the defined criteria, often involving normalization, weighting and scoring to generate actionable insights. Once evaluated, the results are analyzed in the sixth step, "Analyze Results," to understand trade-offs, conflicts and priorities among alternatives. Finally, in the seventh step, "Making a Decision," an informed choice is made by balancing

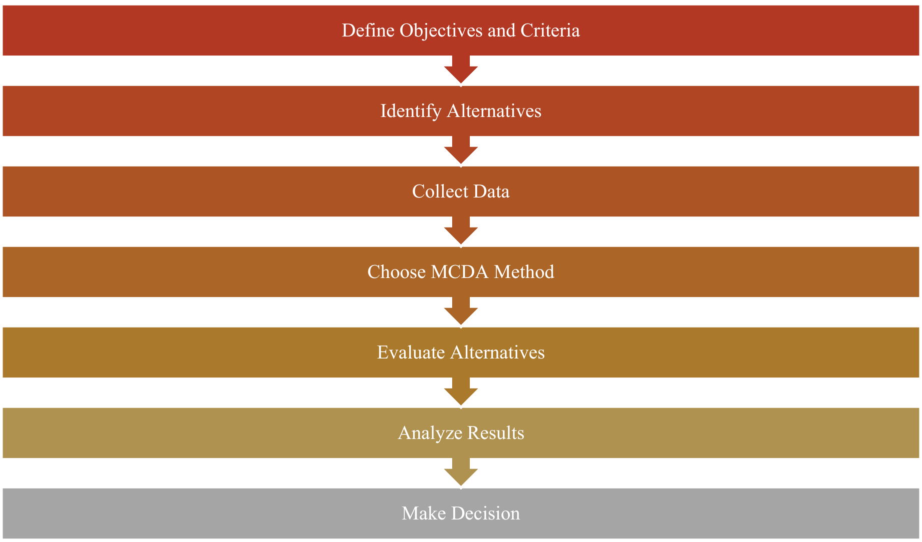

Figure 1. General MCDA framework
Source: Authors' own creation

objective results with expert judgment and practical considerations. The selected alternative should align with the objectives and offer the best overall value.

Cinelli *et al.* (2021) emphasize the escalating significance of MCDA techniques in decision-making, which has consequently fostered the emergence of numerous software tools and libraries encapsulating these methods. Primarily, these tools contribute to formulating the decision issue, enabling decision-makers to systematically and consistently discern relevant criteria and alternatives. Afterward, they simplify the process of appraising alternatives through various techniques. These tools foster the conveyance and visualization of outcomes, equipping decision-makers with coherent and concise data to fortify their decision-making efforts.

Cinelli *et al.* (2014) highlight the rich ecosystem of software tools available in the MCDA sphere. This varied collection includes traditional options such as Expert Choice (Expertchoice, 2020), which embodies the AHP methodology and SuperDecisions (CREATIVE DECISIONS FOUNDATION, 2020), a tool designed to support AHP and the analytic network process (ANP) methodologies. The breadth and diversity of these software offerings underscore the extensive applications of MCDA and the ongoing evolution of decision-making methodologies.

Despite their pervasive utilization, MCDA software tools are not devoid of limitations, resulting in several unaddressed gaps. For example, software such as Expert Choice concentrates solely on a specific method like AHP, constraining users who seek to explore and compare various MCDA techniques for their decision-making challenges. The learning curve associated with some MCDA software tools can be pronounced, necessitating significant time and effort from users to comprehend and master the software's intricacies and capabilities. Moreover, commercial MCDA software tools may carry substantial costs, creating barriers for smaller organizations or individual researchers, and the proprietary nature of closed-source MCDA software can impede collaboration and ongoing refinement. These gaps and challenges underscore the imperative for the continuous advancement and betterment of MCDA software tools, ensuring they effectively cater to the ever-evolving needs of researchers and practitioners.

As mentioned above, an ideal MCDA software should address a variety of considerations. It should support multiple MCDA methods, enabling users to explore and compare diverse techniques while offering customization to adapt the decision-making process to their needs. User-friendliness, intuitive interface and comprehensive documentation should reduce the learning curve. As an open-source platform, the software would foster collaboration and continuous improvement, allowing users to stay abreast of the latest advancements. Seamless integration with other data management, analysis, or visualization tools would simplify incorporating MCDA results into broader workflows.

Large language models (LLM) have significantly advanced natural language processing; these models can generate human-like text, understand context and respond to queries in a highly interactive and engaging manner (KIM *et al.*, 2023). This capability can be integrated into MCDA, revolutionizing how complex decision-making problems are approached. Integrating LLM enables a conversational interface for the decision-making process, which is particularly helpful in explaining the relative weights of different criteria, the outranking of various alternatives and how final decisions are derived. This approach offers a highly interactive and intuitive way to navigate the intricacies of MCDA, making complex decision-making more accessible and understandable.

The marriage of MCDA software functionality and the cutting-edge capabilities of LLM forms a revolutionary approach that streamlines and enhances complex decision-making processes. This innovative amalgamation facilitates an interactive, user-focused platform that bridges the gap between the computational rigidity of MCDA and the dynamic conversational nature of LLM, enlivening the decision-making process in an illuminating and engaging manner. Furthermore, Python emerges as an ideal environment for executing this integration. Renowned for its versatility, user-friendliness and robust open-source community, Python's simplicity, coupled with a vast selection of libraries, allows for smooth integration with a wide array of data management, analysis and visualization tools.

In the context of MCDA software and advancements in LLMs, we introduce pyDecision, a comprehensive and pioneering Python library poised to transform the MCDA domain. Our library equips its users with the capability to tailor an impressive array of 70 distinct MCDA methods to their unique decision-making needs, including widely acclaimed methods such as AHP, TOPSIS and ELimination Et Choix Traduisant laRÉalité (ELECTRE) family, among others. At the heart of pyDecision lies a comprehensive feature suite that empowers users to construct, assess and unravel complex decision-making scenarios with unparalleled depth and clarity.

This article comprehensively explores the pyDecision library, encapsulating its extensive capabilities and unique offerings. Our narrative initiates with an introduction to the environment of MCDA software tools, including the implemented MCDA methods in pyDecision, highlighted in Section 2. Section 3 presents code snippets showcasing the functionalities of pyDecision and its general framework. Section 4 explores the pyDecision library, detailing its capabilities to compare rank generation methods and integration with a LLM, underscoring the synergy of this combination. Section 5 compares methods that can generate weights and their integration with a LLM. Section 6, we evaluate the efficiency and performance of various MCDA techniques in solving decision problems involving a large number of criteria and/or alternatives. Conclusively, Section 7 encapsulates the indelible contributions pyDecision has made to the scientific community.

2. Related software

MCDA, as a structured methodology, enhances decision-making processes by guiding the identification of potential alternatives, the selection of assessment criteria and the consolidation of various stakeholder preferences. This methodology's robustness and utility have been recognized across diverse fields, leading to a substantial increase in dedicated software tools designed to streamline and optimize MCDA processes. (BISDORF *et al.*, 2015; GRECO *et al.*, 2016).

In this context, we adopt a more expansive definition of "software," encompassing standalone applications and various libraries and packages. Libraries and packages are collections of precompiled routines, functions and multiple resources. Libraries are typically organized sets of reusable code and functions that developers can call upon within their programs, enhancing efficiency and reducing redundancy. Packages are similar in function to libraries but often include additional components like documentation, configuration data and other ancillary resources. Both libraries and packages provide a way to encapsulate proven solutions, promoting reusability, maintainability and the collaborative evolution of software (SOMMERVILLE, 2011).

MCDA tools and libraries can be broadly categorized into two primary classes based on their accessibility and adaptability: proprietary and open-source. Each class plays a distinct role in decision-making, balancing usability, transparency and flexibility. Below is an overview of existing MCDA tools categorized under these two classes, and their objectives and limitations are also discussed.

2.1 Proprietary

1000minds (1000MINDS, 2023), a Web-based application, epitomizes this approach by exclusively using the Potentially All Pairwise RanKings of all possible Alternatives method, created by Hansen and Ombler (2008) to cater to decision-making processes. As a proprietary platform, 1000minds mitigates some limitations by providing complimentary access to academic researchers and students, enhancing accessibility within certain academic and research communities. Another example is Expert Choice, a proprietary tool and a specialized decision-making software operationalizing the AHP method. It simplifies AHP implementation but is closed-source, limiting custom extensions or modification opportunities. Criterium DecisionPlus (INFOHARVEST, 2023) is an MCDA-based software for decision-making processes. It prominently implements the AHP and the Simple Multi-Attribute Rating Technique (SMART). D-Sight (ENTERPRISE SOFTWARE, 2023) focuses on decision support software and services. Specifically, D-Sight's tools use the Preference Ranking Organization Method for Enrichment Evaluation (PROMETHEE), Geometrical Analysis for Interactive Decision Aid (GAIA), Multi-Attribute Utility Theory (MAUT) and AHP.

2.2 Open source

VoraciousAHP (PEREIRA, 2022), 3 MO-AHP (FLORIANO *et al.*, 2022), pyMissingAHP (HEYMANN *et al.*, 2024), SuperDecisions, J-Electre (PEREIRA and NEPOMUCENO, 2021) and Electre Tree (MONTENEGRO DE BARROS *et al.*, 2021) collectively represent a range of solutions within the MCDA domain that implement one method or the family of the method and are open source. VaraciousAHP uses the AHP method and can refine the Pairwise Comparison Matrix for consistency. This notion of consistency in AHP is also addressed by 3 MO-AHP, which extends its utility to both traditional AHP and Fuzzy AHP, giving multiple solutions depending on the decision-maker requirements. The pyMissingAHP uses a Genetic Algorithm approach to handle missing values in AHP and Fuzzy AHP. It generates values that ensure the pairwise comparison matrix remains consistent, preserving the integrity of the decision-making process. As a freely available tool, Super Decisions is designed to facilitate learning the AHP and ANP methodologies. In contrast, J-Electre offers a dedicated platform implementing the ELECTRE family of methods, including ELECTRE I, Is, Iv, II, III, IV and Tri, providing versatility across various MCDA scenarios. Finally, Electre Tree, specializing in the Electre Tri algorithm, innovates by inferring missing parameters and accommodating incomplete information without sacrificing robustness. Ranking Trees (RODRIGUES *et al.*, 2021) offers a specialized algorithm capable of eliciting various combinations of parameters for ELECTRE II, II and IV and PROMETHEE I, II, III and IV methods. The Python libraries pyFDM (WIĘCKOWSKI *et al.*, 2023a) and pyIFDM (WIĘCKOWSKI *et al.*, 2023b) provide sophisticated tools for handling uncertain data and intuitionistic fuzzy MCDA methods, respectively. Further extending Python's reach in this area, pymcdm (KIZIELEWICZ *et al.*, 2023) and pyrepo-mcda (Wątróbski *et al.*, 2022) encompass a broad array of MCDA methods. Meanwhile, R MCDA (Bigaret *et al.*, 2017) capitalizes on the statistical prowess of the R environment to deploy various algorithms, and JMcDM (SATMAN *et al.*, 2021) introduces the Julia language to multiple-criteria decision-making with an intuitively designed interface. These open-source libraries, featuring various MCDA methods, contribute to a rich and multifaceted decision-making ecosystem.

Python and R have cultivated strong, vibrant communities around their open-source libraries, making them central to advancing MCDA. Each library for these languages has unique objectives and limitations, supplying diverse user needs while raising collaboration and innovation. pyFDM and pyIFDM, both Python-based, target handling uncertainty in decision-making. pyFDM specializes in fuzzy MCDA methods, making it valuable for applications requiring vagueness or imprecision in criteria evaluation. pyIFDM extends this further into intuitionistic fuzzy logic, accommodating even greater uncertainty. However, both libraries are highly specialized, making them less versatile for users needing a broader range of MCDA methods. More general-purpose Python libraries, such as pymcdm and pyrepo-mcda, aim to cover MCDA techniques comprehensively. pymcdm supports a variety of methods, offering flexibility for decision-makers addressing different types of problems. Similarly, pyrepo-mcda focuses on reproducibility, making it particularly valuable for academic research. Despite their strengths, these Python libraries face challenges in standardization and documentation as they rely heavily on community-driven development.

In the R ecosystem, R MCDA capitalizes on R's strengths in statistical analysis, integrating MCDA techniques with advanced data analytics and visualization tools. This combination makes it a powerful choice for researchers merging statistical workflows with decision analysis. However, R MCDA, like many R packages, often requires a steep learning curve for users unfamiliar with R's syntax and environment, which may deter new adopters. JMcDM, implemented in Julia, prioritizes computational efficiency and user-friendly interfaces. While its focus on modern MCDA methods aligns with cutting-edge research, it has a smaller community than Python and R, potentially limiting resources and contributions.

The comparison between Python and R communities highlights Python's broader versatility and more approachable syntax, making it popular among general users and multidisciplinary teams. R, however, excels in statistical rigor and analysis, which is invaluable for academic and highly technical MCDA applications. Both ecosystems benefit from their open-source nature, fostering transparency and innovation, but Python's wider adoption and developer community make its libraries more accessible and collaborative for broader audiences.

Unlike proprietary models, open-source platforms enable complete transparency, allowing for rigorous academic scrutiny and ongoing improvements. Furthermore, being open-source means that these tools can be customized to meet the unique requirements of any specific project or research work. This adaptability encourages an inclusive, collaborative environment where methods and tools are continually refined, extended and updated by a global community of researchers and practitioners. Moreover, these platforms typically support various MCDA methods, offering extensive coverage and versatile solutions to diverse decision-making scenarios. In contrast, focusing solely on a single method or a specific family of methods can limit the range of problems that can be effectively addressed, restricting adaptability to varying decision contexts.

Therefore, in light of the advantages mentioned earlier, Open Source software that supports various MCDA methods offers a rich tapestry of methodologies, enabling a more robust and adaptable framework for multiple-criteria decision-making. A detailed comparison of these libraries, juxtaposed with pyDecision, which also follows the Open Source approach supporting various MCDA methods, is available in Table 1. In this comparison, an (X) denotes the implementation of the classic method, (F) signifies the implementation of the fuzzy version and (I) represents the incorporation of the intuitionistic fuzzy version. Also, the methods marked with an asterisk (∗) indicate that they only generate weights for each criterion, and the hashtag (#) indicates voting methods adapted to the MCDA context.

Table 1 serves as a comprehensive guide to the scene of MCDA libraries, highlighting their specialized methodologies through a structured coding system. Several libraries have

Table 1. MCDA methods

Methods	AHP (Saaty, 1990)	ARAS (Zavadskas and Turskis, 2010)	#Borda (Borda, 1781)	*BWM (Rezaei, 2015)	*CILOS (Zavadskas and Podvezko, 2016)	CoCoSo (Yazdani et al., 2019)	CODAS (Ghorabaee et al., 2016)	COMET (Sałabun, 2015)	#Copeland (Copeland, 1951)	COPRAS (Zavadskas et al., 1994)	CRADIS (Puška et al., 2022)	*CRITIC (Diakoulaki et al., 1995)	DEMATEL (Fontela and Gabus, 1976)	EDAS (Ghorabaee et al., 2015)	ELECTRE (Roy, 1968)	*Entropy (Shannon, 1948)	GRA (Deng, 1989)
pyDecision	XF	XF	X	XF	X	X	X		X	XF	X	X	XF	XF	X	X	X
JMcDM	X	X				X	X		X	X		X	X	X	X	X	X
pyFDM		F					F			F				F			
pyIFDM		I					I			I				I			
pymcdm		X				X	X	X		X				X			
pyrepo-mcda	X	X				X	X			X	X			X			
R MCDA	X														X		
	*IDOCRIW (Zavadskas and Podvezko, 2016)	MABAC (Pamučar and Ćirović, 2015)	MACBETH (Costa and Vansnick, 1994)	MAIRCA (Pamučar et al., 2018)	MARCOS (Stević et al., 2020)	MARE (Hodgett et al., 2014)	MAUT (Keeney and Raiffa, 1976)	*MEREC (Keshavarz-Ghorabaee et al., 2021)	MOORA (Brauers and Zavadskas, 2006)	MOOSRA (Jagadish and Ray, 2014)	MRSORT (Bouyssou and Marchant, 2007)	MULTIMOORA (Brauers and Zavadskas, 2010)	OCRA (Miloš et al., 2015)	ORESTE (Roubens, 1982)	PIV (Mufazzal and Muzakkir, 2018)	PIROMETHEE (Brans and Vincke, 1985)	PROSA-C (Ziemba, 2020)
pyDecision	X	X	X	X	X		X	X	XF	X		X	XF	X	X	X	
JMcDM		X		X	X			X	X						X	X	
pyFDM		F		F					F				F				
pyIFDM		I		I					I								
pymcdm		X		X	X				X				X			X	
pyrepo-mcda		X			X							X				X	X
R MCDA						X					X					X	
	PSI (Maniya and Bhatt, 2010)	Regime (Hinloopen and Nijkamp, 1990)	ROV (Madić et al., 2016)	SAW (Churchman et al., 1954)	SECA Keshavarz-Ghorabaee et al., 2018)	SMART (Edwards, 1971)	SPOTIS (Dezert et al., 2020)	SRMP (Khannoussi et al., 2022)	SURE (Hodgett and Siraj, 2019)	TODIM (Gomes and Rangel, 2009)	TOPSIS (Hwang and Yoon, 1981)	UTA (Jacquet-Lagreze and Siskos, 1982)	VIKOR (Opricović, 1990)	WINGS (Michnik, 2013)	WSM (Fishburn, 1967)	WPM (Miller and Starr, 1960)	WASPAS (Zavadskas et al., 2014)
pyDecision	X	X	X	X		X	X			X	XF		XF	X	XF	XF	XF
JMcDM	X		X	X	X						X		X			X	X
pyFDM											F		F				
pyIFDM											I		I				
pymcdm							X				X		X				
pyrepo-mcda				X			X				X		X				X
R MCDA								X	X		X	X					

Source(s): Authors' own creation

emerged to support a variety of methodological approaches. Classic MCDA methods like AHP, additive ratio assessment (ARAS) and simple additive weighting (SAW) remain universally popular across libraries. This universal adoption serves multiple purposes: it offers a reliable foundation for decision-making, acts as an accessible entry point for those new to the field and provides a rigorous benchmark against which to measure emerging methods. Fuzzy adaptations of these conventional methods are also included, such as Fuzzy ARAS (FU *et al.*, 2021), Fuzzy complex proportional assessment (COPRAS) (NARANG *et al.*, 2021) and Fuzzy evaluation based on distance from average solution (EDAS) (ZINDANI *et al.*, 2019). These fuzzy variants extend the applicability of MCDA techniques to situations demanding more nuanced treatment of uncertainty, thereby broadening the decision-making toolkit available to practitioners.

Weight generation methods like Best-Worst method (BWM) [including its simplified version – Amiri *et al.* (2021)], compromise ideal list of solutions (CILOS) and criteria importance through intercriteria correlation (CRITIC) are crucial, as they set the stage for more complex evaluations. Libraries progressively acknowledge the importance of these weight assignments, underscoring their fundamental role in the accuracy of the overall analysis. Finally, including voting methods such as Borda and Copeland marks an interdisciplinary approach to MCDA, integrating techniques from social choice theory to enrich the decision-making frameworks. Libraries like pyDecision and JMcDM have embraced these methods, reflecting the adaptability of MCDA to incorporate various strategies from different disciplines for more robust decision support. Collectively, these trends signal an MCDA community committed to methodological diversity, foundational rigor and adaptive innovation to meet the complexities of real-world decision-making.

pyDecision is a comprehensive and adaptable tool within the scope of MCDA. Not only does it support an extensive 70 different MCDA methods, but it also spans a wide gamut from established classics like AHP, ELECTRE (I, I-s, I-v, II, III, IV and Tri) and PROMETHEE (I, II, III, IV, V, VI and GAIA), to innovative, modern approaches such as compromise ranking of alternatives from distance to ideal solution (CRADIS) and entropy-critic promethee (EC PROMETHEE) (BASILIO *et al.*, 2023). This diversity makes it versatile enough to address various decision-making challenges. Furthermore, pyDecision extends its reach by offering fuzzy logic versions of some methods, making it a go-to option for scenarios demanding a more nuanced treatment of uncertainty and ambiguity. It also incorporates advanced criterion-weighting methods like BWM and CRITIC and voting algorithms like Borda and Copeland, providing a rich toolkit for complex decision-making processes. What amplifies pyDecision's utility is its integration of a LLM, specifically ChatGPT (OPENAI, 2023), which allows users to compare results – such as ranks, fuzzy ranks and weights – generated by various methods. This feature enhances the platform's interactivity and simplifies the intricacies involved in multicriteria decisions. With its broad methodological scope and sophisticated features like LLM, pyDecision tries to serve as a one-stop-shop for individuals and organizations confronting diverse decision-making tasks.

JMcDM emerges as a solid contender focusing on classic and well-established methodologies. It excels in its coverage of time-tested algorithms, including but not limited to AHP, ARAS, ELECTRE I and PROMETHEE II, further underscoring its credentials as a versatile tool for classical decision-making scenarios. The library also offers the added benefit of criterion weighting techniques like CRITIC and Entropy, thus, enhancing its flexibility. Notably, however, it does not support fuzzy methods, which limits its utility in scenarios demanding a nuanced approach to uncertainty or ambiguity. JMcDM is a robust choice for those who prioritize classical, well-tested methods.

pyFDM and pyIFDM distinguish themselves through a laser-focused commitment to fuzzy and intuitionistic fuzzy methods, respectively, providing nuanced alternatives to classic techniques like ARAS and EDAS. Both libraries are invaluable for complex decision-making scenarios that require intricate handling of uncertainty, vagueness or ambiguity. pyFDM is particularly adept at tackling problems where the limitations of conventional crisp logic become apparent, offering a more nuanced fuzzy approach to managing uncertainty. Conversely, pyIFDM goes further into the specialized realm of intuitionistic fuzzy methods, offering a sophisticated framework for grappling with even more complex forms of uncertainty and ambiguity. While this high level of specialization equips them to excel in their respective niches, it also restricts their applicability to more generalized or classic MCDA tasks. Thus, pyFDM and pyIFDM serve as potent, specialized tools for specific complex scenarios. Still, their narrow methodological focus may not cater to those needing a more comprehensive suite of decision-making instruments.

pymcdm carves out a distinct space as a middle-of-the-road option emphasizing time-tested, classic methods like PROMETHEE II and its unique offering, the characteristic objects method (COMET) technique. This focused commitment to classic methodologies serves as both its strength and limitation. While it facilitates ease of use and offers a simplified user experience, it also constrains the library's versatility in handling more MCDA problems.

pyrepo-mcda emerges as a balanced, mid-tier contender. What distinguishes it is a well-curated selection of classic methods, such as PROMETHEE II, punctuated by specialized offerings like PROSA-C. These unique inclusions give pyrepo-mcda a distinct edge for those interested in exploring beyond the usual staples. Nevertheless, the library has limitations, particularly its lack of support for fuzzy techniques, which narrows its applicability for complex scenarios demanding nuanced handling of uncertainty or vagueness.

R MCDA offers specialized algorithms such as stochastic robust multi-criteria acceptability analysis preference (SRMP), simple additive ranking using exponential functions (SURE) and multi-attribute ranking exploitation (MARE). R MCDA may not provide an expansive palette of MCDA techniques. Nevertheless, its integration with the R ecosystem and its discerning focus on a handful of reputable methods positions R MCDA as an attractive choice for users grappling with specialized decision-making challenges.

The panorama of MCDA libraries is rich and varied, each distinguished by its strengths, limitations and specializations. The selection caters to different types of users and problem-specific requirements, ranging from the comprehensiveness of pyDecision to the niche focus of libraries like pyFDM and pyIFDM. Regarding method coverage, specific methods like AHP, ARAS, ELECTRE and SAW enjoy widespread support. In contrast, more specialized or less commonly used methods like MARE, simultaneous evaluation of criteria and alternatives (SECA) and SRMP are exclusive to specific libraries. This hints at their specialized nature or limited application in the MCDA field. Implication-wise, pyDecision and JMcDM are all-encompassing choices for generalists, covering various classic and fuzzy methods. Specialists might find libraries like R MCDA, pyFDM and pyIFDM more aligned with their needs, focusing on specific subsets of MCDA methods. However, it is crucial to acknowledge the limitations inherent in existing MCDA software. The predilection for single-model approaches can limit versatility, adapting to a broad spectrum of decision-making scenarios can be challenging, and user-friendliness remains an obstacle, particularly for MCDA novices. Therefore, while each library brings its unique advantages to the table, the choice of the most appropriate tool should be carefully considered based on the specific needs and constraints of the decision-making environment.

3. PyDecision – Overview

In this section, we will offer a concise glimpse into the capabilities of pyDecision by walking through simple code examples. We aim to illustrate how to apply the library's MCDA algorithms once pyDecision is installed and ready.

Furthermore, to make the library accessible to a wider audience, each method in pyDecision is accompanied by a dedicated Colab demo, available through the library's page (https://bit.ly/3tLFGtH). These demos provide interactive, browser-based environments where users can execute code without installing Python or setting up a local development environment. This functionality is particularly advantageous as it works on any device with internet access, such as mobile phones and tablets, with the computational workload handled entirely by Colab servers. In addition, as each method may require specific parameters and setups, these demos ensure clarity and ease of experimentation for users of all expertise levels. For those interested in the theoretical foundations of the methods, each Colab demo is linked to a corresponding scientific paper. These papers explain the methodologies in-depth, discussing their underlying principles, advantages, characteristics and limitations.

Installing pyDecision is straightforward and user-friendly. Since the library is hosted on the Python Package Index (PyPI), you can open a terminal or command prompt in your Python environment and run the command shown in Figure 2.

Once the installation is complete, pyDecision and its required dependencies will be ready for use, allowing the user to explore the library's array of classical and fuzzy MCDA techniques and its AI-driven functionalities immediately. To demonstrate its functionality, we will show how to use pyDecision to solve a generic MCDA problem using the TOPSIS method, a popular MCDA technique. It ranks alternatives based on their proximity to an ideal solution while being farthest from a negative ideal solution.

In Figure 3, we import the necessary libraries to perform the analysis. The NumPy library is imported as "np" and is used to facilitate numerical computations, such as handling the decision matrix as a structured array. From the pyDecision library, we import the "topsis_method" function, which implements the TOPSIS. This method will analyze the data set, calculate relative closeness scores and rank the alternatives based on their proximity to the ideal solution.

The input in Figure 4 shows the "data set," a two-dimensional matrix where each row represents an alternative (e.g. different options to evaluate), and each column corresponds to a criterion (e.g. different factors influencing the decision). Therefore, this data set contains 10 alternatives, each evaluated against four criteria. The numerical values indicate each alternative's performance under each criterion. The "weights" specify the relative importance of each criterion in the decision-making process; here, each weight is set to 0.25,

```
pip install pydecision
```

Figure 2. Installation
Source: Authors' own creation

```
import numpy as np
from pyDecision.algorithm import topsis_method
```

Figure 3. Importing libraries
Source: Authors' own creation

```
dataset = np.array([
                    [6, 8, 4, 7],
                    [9, 3, 4, 6],
                    [4, 9, 7, 3],
                    [8, 2, 5, 8],
                    [4, 9, 2, 3],
                    [7, 5, 9, 9],
                    [9, 6, 3, 1],
                    [3, 5, 7, 6],
                    [5, 3, 8, 5],
                    [4, 6, 3, 8]
                  ])

weights        = [ 0.25,  0.25,  0.25,  0.25]
criterion_type = ['max', 'max', 'max', 'max']
```

Figure 4. Input
Source: Authors' own creation

indicating that all criteria are equally important. The "criterion_type" defines the orientation of each criterion: whether a higher value is desirable (max) or undesirable (min). In this example, all criteria are labeled as max, meaning higher values for each criterion are considered better. This setup reflects scenarios where alternatives are evaluated on positively oriented metrics, such as profit, performance or efficiency, making them straightforward to compare within the decision-making framework.

In Figure 5, the function "topsis_method" takes four main arguments: the "data set," "weights," "criterion_type" and optional parameters "graph" and "verbose." When set to True, the optional "graph" parameter generates a visual ranking of alternatives. At the same time, the "verbose" parameter prints detailed scores for each alternative in the console when also set to True. The function returns an array of "relative closeness" values, where each value indicates how close a given alternative is to the ideal solution, with higher values signifying more favorable alternatives.

The result shown in Figure 6 illustrates the ranking of alternatives based on their relative closeness scores calculated using the TOPSIS method. Each alternative is arranged vertically from the most favorable (top) to the least favorable (bottom). The scores indicate the closeness of each alternative to the ideal solution, where higher values signify better performance. Alternative a6 ranks the highest with a relative closeness score of 0.73, making it the most favorable choice. This is followed by a1 (0.59), a3 (0.53), a4 (0.52) and a8 (0.50). Alternatives a9 and a10 both score 0.49, sharing a similar rank. Further down the list are a2 (0.48) and a5 (0.41), while a7 has the lowest score of 0.40, making it the least favorable alternative. This visual and numerical representation provides a clear understanding of the relative performance of each option, aiding decision-makers in selecting the best alternative.

One of the key advantages of using a Python-based library like pyDecision is its seamless integration with other Python tools and libraries, enabling users to perform advanced analyses beyond the library's built-in capabilities. For instance, sensitivity analysis – critical

```
relative_closeness = topsis_method(dataset, weights, criterion_type, graph = True, verbose = True)
```

Figure 5. TOPSIS
Source: Authors' own creation

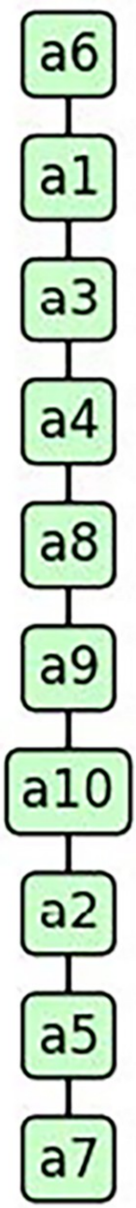

Figure 6. TOPSIS results
Source: Authors' own creation

for understanding how changes in input parameters (e.g. weights) affect decision outcomes – can be easily implemented by combining pyDecision with other Python libraries such as NumPy, Pandas and Matplotlib. This flexibility allows users to explore, visualize and quantify the stability of their rankings under varying conditions, enhancing the robustness of their decision-making processes.

In Figure 7, the code performed a sensitivity analysis using the TOPSIS method by varying the weights assigned to the criteria and analyzing how the ranking of alternatives changes across different weight sets. Seven different weight sets are defined, each distributing the importance of the criteria differently. The TOPSIS method is applied for each weight set to calculate the closeness scores for all alternatives. The alternatives are then ranked in descending order of their closeness scores. The results are stored in a dictionary where each key corresponds to a weight set, and the value is the ranking of alternatives for that set. Finally, the rankings for all weight sets are printed in a format that clearly shows how varying weights influence the order of alternatives.

The sensitivity analysis results, Figure 8, reveal how the rankings of alternatives change as the weights assigned to the criteria are varied. Across all weight sets, alternative a6

```
import numpy as np
from pyDecision.algorithm import topsis_method

dataset = np.array([
                    [6, 8, 4, 7],
                    [9, 3, 4, 6],
                    [4, 9, 7, 3],
                    [8, 2, 5, 8],
                    [4, 9, 2, 3],
                    [7, 5, 9, 9],
                    [9, 6, 3, 1],
                    [3, 5, 7, 6],
                    [5, 3, 8, 5],
                    [4, 6, 3, 8]
                   ])

criterion_type = ['max', 'max', 'max', 'max']

weight_sets    = [
                  [0.25, 0.25, 0.25, 0.25], # Weight Set 1
                  [0.40, 0.20, 0.20, 0.20], # Weight Set 2
                  [0.10, 0.30, 0.30, 0.30], # Weight Set 3
                  [0.30, 0.30, 0.20, 0.20], # Weight Set 4
                  [0.50, 0.10, 0.20, 0.20], # Weight Set 5
                  [0.20, 0.20, 0.30, 0.30], # Weight Set 6
                  [0.10, 0.40, 0.20, 0.30]  # Weight Set 7
                 ]

results = { }

for i, weights in enumerate(weight_sets, 1):
   closeness_scores        = topsis_method(dataset, weights, criterion_type, graph = True, verbose = True)
   ranked_alternatives     = np.argsort(closeness_scores)[::-1]
   ranked_alternatives     = ['a' + str(i + 1) for i in ranked_alternatives]
   results[f'Weight Set {i}'] = ', '.join(ranked_alternatives)

for weight_set, ranking in results.items():
   print(f'{weight_set} -> {ranking}')
```

Figure 7. Sensitivity analysis
Source: Authors' own creation

```
Weight Set 1 -> a6, a1, a3, a4, a8, a9, a10, a2, a5, a7
Weight Set 2 -> a6, a2, a4, a1, a7, a9, a3, a10, a8, a5
Weight Set 3 -> a6, a1, a3, a8, a10, a9, a4, a5, a2, a7
Weight Set 4 -> a6, a1, a3, a2, a7, a4, a5, a10, a8, a9
Weight Set 5 -> a6, a4, a2, a7, a1, a9, a10, a3, a8, a5
Weight Set 6 -> a6, a4, a1, a8, a9, a3, a10, a2, a5, a7
Weight Set 7 -> a1, a6, a3, a10, a5, a8, a4, a9, a2, a7
```

Figure 8. Sensitivity analysis result
Source: Authors' own creation

consistently ranks either first or second, indicating it is a robustly favorable option regardless of the weighting scheme. Similarly, alternatives such as a1, a3 and a4 frequently appear near the top of the rankings, suggesting they are generally competitive choices. On the other hand, alternatives like a7, a5 and a2 tend to rank lower across most weight sets, implying they are less favorable under the given scenarios. The variability in the middle rankings (e.g. a9, a10 and a8) highlights these alternatives' sensitivity to weight distribution changes, demonstrating how their performance relative to others depends significantly on the importance placed on specific criteria.

In Figure 9, the code snippet demonstrates the use of the method to calculate objective weights for criteria in a decision-making problem. The input consists of a "data set," where each row corresponds to an alternative, and each column represents a criterion (five alternatives evaluated across four criteria). The "criterion_type" specifies whether each criterion is to be minimized (min) or maximized (max), reflecting the decision-maker's preferences. The "critic_method" function computes the weights for each criterion based on their contrast intensity (variation) and correlation with other criteria. These weights objectively represent the relative importance of each criterion derived entirely from the data set. The output is the calculated weights for all criteria. The code iterates through these weights, printing each one rounded to three decimal places, labeled as w(g1), w(g2), etc. where g1, g2 and so on correspond to the criteria.

Figure 10 indicates the relative importance of each criterion as determined by the CRITIC method. Criterion g2 has the highest weight (0.284), suggesting it has the greatest influence

```
import numpy as np
from pyDecision.algorithm import critic_method

dataset = np.array([
                    [250, 16, 12, 5],
                    [200, 16,  8, 3],
                    [300, 32, 16, 4],
                    [275, 32,  8, 4],
                    [225, 16, 16, 2]
                  ])

criterion_type = ['min', 'max', 'max', 'max']

weights = critic_method(dataset, criterion_type)

for i in range(0, weights.shape[0]):
  print('w(g'+str(i+1)+'): ', round(weights[i], 3))
```

Figure 9. CRITIC
Source: Authors' own creation

```
w(g1): 0.270
w(g2): 0.284
w(g3): 0.255
w(g4): 0.191
```

Figure 10. CRITIC results
Source: Authors' own creation

on decision-making. Criterion g1 follows with a weight of 0.270, indicating it also plays a significant role but is slightly less impactful than g2. Criterion g3 has a moderate weight of 0.255, contributing meaningfully to the decision but with less emphasis than the first two criteria. Finally, criterion g4 has the lowest weight (0.191), indicating it has the least influence on decision-making. These weights provide an objective perspective on the importance of each criterion, derived directly from the data set's inherent structure and relationships.

The code in Figure 11 demonstrates how integrating external Python libraries, such as Matplotlib, can complement and enhance the capabilities of pyDecision. The weights are visualized as a bar chart, where each bar represents a criterion, and its height corresponds to its computed weight. Text annotations are added to display the exact weight values, and the plot includes titles, labels and a grid for better interpretability.

In Figure 12, the bar chart illustrates the criteria weights computed using the CRITIC method, where each criterion is represented by a bar. These weights objectively reflect the importance of each criterion based on the data set's structure, with higher weights indicating

```
import matplotlib.pyplot as plt
import numpy as np
from pyDecision.algorithm import critic_method

dataset = np.array([
                    [250, 16, 12, 5],
                    [200, 16,  8, 3],
                    [300, 32, 16, 4],
                    [275, 32,  8, 4],
                    [225, 16, 16, 2]
                   ])

criterion_type = ['min', 'max', 'max', 'max']
weights        = critic_method(dataset, criterion_type)
criteria       = ['g1', 'g2', 'g3', 'g4']

plt.figure(figsize = (8, 6))
bars = plt.bar(criteria, weights, color = 'skyblue', edgecolor = 'black')
for bar in bars:
   plt.text(bar.get_x() + bar.get_width() / 2, bar.get_height() - 0.02, f"{bar.get_height():.3f}", ha = 'center', va =
'bottom', fontsize = 12, color = 'black')
plt.title('CRITIC Method: Criterion Weights', fontsize = 16)
plt.xlabel('Criteria',  fontsize = 14)
plt.ylabel('Weights', fontsize = 14)
plt.ylim(0, 0.35)
plt.grid(axis = 'y', linestyle = '--', alpha = 0.7)
plt.tight_layout()
```

Figure 11. Graph code
Source: Authors' own creation

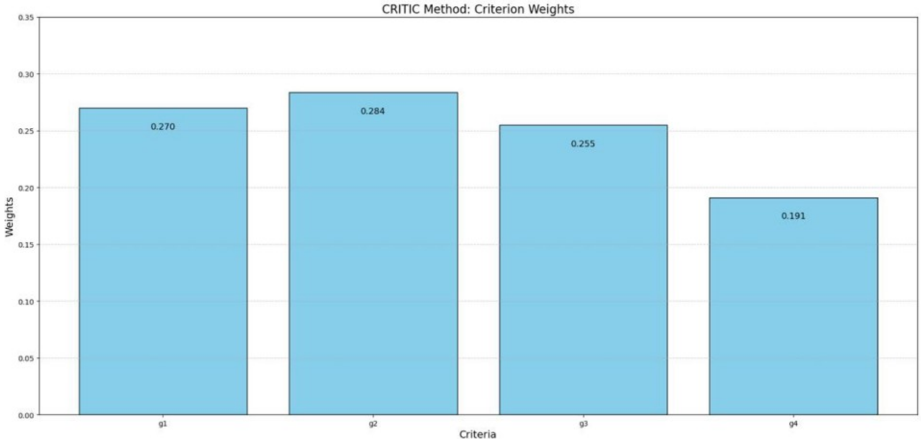

Figure 12. Graph code result
Source: Authors' own creation

criteria that are more critical to the evaluation process. The visualization provides an intuitive understanding of the relative importance of each criterion.

This section aimed to highlight some of the core functionalities of pyDecision, including its ability to perform sensitivity analysis and visualization while integrating with other Python libraries, demonstrating that its flexibility extends beyond the tools provided within pyDecision itself. In the upcoming sections, we will focus on case studies that show the integration of LLMs within pyDecision. These case studies will explore how LLMs enhance decision-making by providing natural language interpretations, explaining results and comparing rankings.

4. PyDecision – large language model – first case study

In this section, we aim to demonstrate the versatility of pyDecision in conjunction with ChatGPT. The first case study explores various methods capable of generating rankings, including two methods from the academic literature not natively supported by pyDecision. ChatGPT plays an integral role in interpreting the outcomes. We use a sequence of targeted prompts to guide the decision-maker in evaluating and understanding the results. One can note that the underlying analysis would essentially be the same since fuzzy methods produce crisp rankings even with fuzzy input. Therefore, we will not conduct a separate evaluation for ranks generated by fuzzy methods to avoid redundancy.

The first case study can be accessed at < https://bit.ly/47aAVrZ >.

4.1 First case study – methods comparison

The first case study involves selecting an optimal material for a cryogenic storage tank destined to transport liquid nitrogen. The specific performance requirements of the storage tank were meticulously translated into corresponding material attributes. The chosen material must demonstrate excellent weldability, processability, lower density and specific heat metrics, among other essential qualities. It must also present a minimal thermal expansion coefficient and thermal conductivity while maintaining sufficient strength, stiffness and toughness at the operating temperature. For this particular application, materials showcasing higher mechanical properties are sought after, with emphasis on the highest values in toughness index (TI), yield strength (YS) and Young's modulus (YM). The selection process evaluates seven distinct attributes, representing the desired material properties against seven alternative materials. Among these, TI, YS and YM are beneficial attributes (max), where higher values are advantageous. Conversely, the remaining four attributes – namely, density (D), thermal expansion (TE), thermal conductivity (TC) and specific heat (SH) – are deemed nonbeneficial (min), as their lower values align more closely with the application's needs (RAO, 2006; Dehghan-Manshadi *et al.*, 2007).

The alternatives considered for the material selection, as outlined in Table 2, include a1 (Material 1 – Al 2024-T6), a2 (Material 2 – Al 5052-O), a3 (Material 3 – SS 301-FH), a4 (Material 4 – SS 310-3AH), a5 (Material 5 – Ti-6Al-4V), a6 (Material 6 – Inconel 718) and a7 (Material 7 – 70Cu-30Zn). To tailor the selection process to the application's specific needs, Table 2 also shows the weights Rao (2006) and Manshadi *et al.* (2007) allocated to reflect each attribute's relative importance. These weights function as pivotal parameters in assessing each material's suitability, steering the decision-making toward alignment with the application's distinct requirements.

The data set, as displayed in Table 2, reveals the complexity of the decision-making process, as no single material excels across all the evaluated attributes. This intricate balance of competing properties underscores the challenge of material selection for this highly specialized task.

Table 3 presents the rankings of each alternative as determined by two distinct techniques: the Graph Theory technique, as proposed by Rao (2006), and a Nonlinear Normalization method enhanced by a modified digital logic approach, as described by Manshadi *et al.* (2007).

Regarding the ranks for each alternative, Table 3 shows appealing convergences and divergences. For example, both methods unanimously ranked the third alternative as the best (first place); the same goes for the second and fifth alternatives. However, there are variations in the middle rankings; the first alternative was ranked sixth according to Rao (2006) but fared slightly better at fifth place in Manshadi *et al.* (2007).

Table 4 is structured to display a comprehensive comparison of various MCDA methods. The rows list the methods under consideration. The columns represent the different alternatives being evaluated. Within the cells, the rank is assigned for each alternative. The following methods were used: ARAS, combined compromise solution (CoCoSo), combinative distance-based assessment (CODAS), COPRAS, CRADIS, EDAS, grey relational analysis (GRA),

Table 2. First case study

Alternatives	TI (MAX)	YS (MAX)	YM (MAX)	D (MIN)	TE (MIN)	TC (MIN)	SH (MAX)
Weights	0.28	0.14	0.05	0.24	0.19	0.05	0.05
a1	75.50	420	74.2	2.80	21.40	0.370	0.16
a2	95	91	70	2.68	22.10	0.330	0.16
a3	770	1,365	189	7.90	16.90	0.040	0.08
a4	187	1,120	210	7.90	14.40	0.030	0.08
a5	179	875	112	4.43	9.40	0.016	0.09
a6	239	1,190	217	8.51	11.50	0.310	0.07
a7	273	200	112	8.53	19.90	0.290	0.06

Source(s): Authors' own creation

Table 3. First case study results

Alternatives	Rao (2006)	Manshadi *et al.* (2007)
a1	17.29 (6th)	−1.71 (5th)
a2	16.26 (7th)	−8.75 (7th)
a3	39.11 (1st)	47.40 (1st)
a4	30.63 (3rd)	31.88 (4th)
a5	34.05 (2nd)	43.52 (2nd)
a6	29.04 (4th)	33.44 (3rd)
a7	20.04 (5th)	−3.07 (6th)

Source(s): Authors' own creation

Table 4. Rank comparison

Methods	a1	a2	a3	a4	a5	a6	a7		a1	a2	a3	a4	a5	a6	a7
Rao (2006)	6	7	1	3	2	4	5	OCRA	6	7	1	2	3	4	5
Manshadi *et al.* (2007)	5	7	1	4	2	3	6	ORESTE	5	6	2	3	1	4	7
ARAS	5	6	1	4	2	3	7	PIV	5	6	1	4	2	3	7
CoCoSo	5	6	1	3	2	4	7	PROMETHEE II	7	6	1	4	2	3	5
CODAS	4	2	1	6	3	5	7	PROMETHEE IV	6	7	1	3	4	2	5
COPRAS	5	7	1	4	2	3	6	EC PROMETHEE	5	7	1	3	2	4	6
CRADIS	6	7	2	3	1	4	5	PSI	5	6	2	4	1	3	7
EDAS	6	7	1	4	2	3	5	ROV	5	6	1	4	2	3	7
GRA	5	4	1	6	2	3	7	SAW	5	6	1	4	2	3	7
MABAC	5	6	1	3	2	4	7	SPOTIS	5	6	1	4	2	3	7
MACBETH	5	6	1	4	2	3	7	TODIM	5	6	1	3	2	4	7
MAIRCA	5	6	1	4	2	3	7	TOPSIS	5	6	1	4	2	3	7
MARCOS	5	6	1	4	2	3	7	VIKOR	3	5	6	4	1	2	7
MAUT	4	6	1	3	5	2	7	WSM	5	6	1	4	2	3	7
MOORA	5	6	1	4	2	3	7	WPM	5	6	1	3	2	4	7
MOOSRA	5	7	1	4	2	3	6	WASPAS	5	6	1	3	2	4	7
MULTIMOORA	5	7	1	3	2	4	6	-//-	–	–	–	–	–	–	–

Source(s): Authors' own creation

multi-attributive border approximation area comparison (MABAC), measuring attractiveness by a categorical based evaluation technique (MACBETH), multi-attribute ideal-real comparative analysis (MAIRCA), measurement of alternatives and ranking according to compromise solution (MARCOS), MAUT, multi-objective optimization on the basis of ratio analysis (MOORA), multi-objective optimization on the basis of simple ratio analysis (MOOSRA), MOORA plus full multiplicative form (MULTIMOORA), ordinal classification and ranking analysis (OCRA), organisation, rangement et synthese de donnees relationnelles (ORESTE), preference index value (PIV), PROMETHEE II, PROMETHEE IV, EC PROMETHEE, preference selection index (PSI), range of value (ROV), SAW, stable preference ordering towards ideal solution (SPOTIS), Tomada de Decisão Iterativa Multicritério (TODIM), TOPSIS, VIKOR, weighted sum model (WSM), weighted product model (WPM) and weighted aggregated sum product assessment (WASPAS). In addition, the results obtained by Rao (2006) and Manshadi *et al.* (2007) were included.

Specific parameters are relevant to particular methods, as follows: for PROMETHEE II, PROMETHEE IV or EC PROMETHEE: Q = 5, 10, 1.7, 0.02, 0.01, 0.01 and 0.01; S = 7, 15, 1.9, 0.03, 0.02, 0.02 and 0.02; p = 9, 20, 2.5, 0.04, 0.03, 0.03 and 0.03; F = Usual Preference. For EC PROMETHEE additionally: custom sets = 0.5, 0.5, 0.5, 0.5, 0.5, 0.5 and 0.5; Iterations = 10,000. For MAUT: utility functions = exponential/step functions (max/min criterion); Step Size = 1. For CoCoSo: L = 0.5. For CODAS: Lambda = 0.02. For GRA: Epsilon = 0.5. For ORESTE: Alpha = 0.4. For TODIM: Teta = 1. For VIKOR: strategy coefficient = 1. For SPOTIS: S_{min} = 70, 90, 50, 2.7, 9.0, 0.01 and 0.05; S_{max} = 780, 1,200, 220, 9.7, 24.0, 0.41 and 0.25.

Upon analyzing Table 4, several patterns and variations can be observed. The alternative a3 has the most stable rank across almost all methods. It is frequently ranked at Position 1, indicating a strong consensus that it is the best alternative according to the evaluated criteria. Alternatives a2 and a7 show a moderate variation in their ranks; while they are generally ranked towards the middle or bottom across methods, the exact position fluctuates. For instance, for a2, rankings range from 2 (in CODAS) to 7 (in several methods). Conversely, in the alternatives a4 and a5, the rank varies extensively among the methods. For example, a4 is ranked anywhere from third to sixth place, and a5 moves between the first and fourth positions, indicating less consensus on assessing these alternatives.

Should the decision-maker desire to do so, they can derive a final rank using the predominant ranks for each alternative. However, caution is advised in multimodal scenarios, where an alternative may possess multiple ranks, or in tie situations where different alternatives share identical ranks. In our case study, the final rank obtained by the mode is a3, a5, a6, a4, a1, a2 and a7. Alternatively, a voting system can be used to ascertain ranks. Borda and Copeland's methods are natively available in pyDecision. For this case study, both methods computed the same final rank as the mode. Nevertheless, the decision-maker retains the flexibility to use any other preferred method.

4.2 First case study – methods comparison – large language model aid

To extract deeper insights and potentially uncover hidden patterns in Table 4, we'll leverage the analytical capabilities with ChatGPT (model gpt4). We plan to use a sequence of targeted prompts to augment our initial analysis by shedding light on aspects we might have overlooked.

Prompt 1: Which methods are more similar and which ones are more dissimilar?

Response: The similarity of methods can be analyzed by looking at how consistently they rank the alternatives. From the given outranking table, it is not possible to declare precisely which methods are more similar and which are more dissimilar without additional calculations (e.g. calculating Euclidean distance or Pearson's correlation). However, some general trends can be observed. For instance, within the methods, "aras" and "macbeth," the provided ranking values in the table appear to show a high degree of similarity across all alternatives as they both rank alternatives the same way. In the same way, "mabac" and "marcos" or "maut" and "moora" methods seem to give similar rankings across the alternatives. On the other hand, a method like "vikor" appears to be more dissimilar from others. For example, in "vikor," the first and fifth alternatives are given a rank of sixth and first, which differs significantly from most other methods. Nevertheless, this is a preliminary assessment, and a more accurate interpretation would necessitate mathematical computations such as calculating correlation coefficients to determine the extent of similarity or dissimilarity. This also assumes that the ordering of the alternatives is all that matters, and not the actual values attached to each position. [sic]

Prompt 2: Which alternative(s) consistently ranks high or low across all the methods?

Response: The alternative that consistently ranks high across all the methods is the third one. It ranks first in almost all methods except two (rov and oreste), where it ranks second.

On the other hand, the alternative that consistently ranks low across all the methods is the sixth one. It mostly ranks seventh, which is the lowest, with a few exceptions where it ranks slightly higher. [sic]

Prompt 3: Are there any noticeable differences in rankings across the methods? if so, what could account for these differences?

Response: Yes, there are noticeable differences in rankings across the MCDA methods. The third alternative, for instance, consistently ranks first in almost all MCDA methods – except for a few like OCRA, PIV and SAW where it ranks second or lasts sixth in VIKOR – indicating a strong performance across multiple evaluation standards. The sixth alternative, on the other hand, mostly ranks in the middle. Still, there are also few methods like MOORA, TODIM and COPRAS where it achieves weaker positions, and a better position in EC_Promethee and WASPAS. The same goes for the other alternatives, where one can notice variations in their positions depending on the applied MCDA method. These discrepancies may be due to different prioritization and analysis mechanisms intrinsic to each MCDA method. Depending on the method, various aspects such as weight assignment, type of optimization (Max or Min), preference analysis, consideration of subjective opinions, normalization methods, etc. could have different impacts on the ranking. For instance, some methods may assign more weight to certain criteria, or directly incorporate the decision-maker's preferences into the decision-making process, thus, influencing the final outcomes. Others could use an additive aggregation function, while others could use a multiplicative aggregation function, leading to alternate outcomes. In other words, the specific methodology of each MCDA technique and their differing treatment of both the criteria and the alternatives' performances on these criteria could result in different rankings. In addition, uncertainty in original data, potential biases in data collection and treatment tools and subjectivity in interpreting the criteria could also contribute to differences in the final ranking. Moreover, method-related factors, such as sensitivity to changes in parameters, handling of inequality among the criteria and the ability to consider interactions among criteria – along with problem-related aspects, such as complexity of the problem, number and nature of alternatives and criteria and conflict levels among criteria, could also play a role. [sic]

Prompt 4: Is there a consensus among the methods for any specific alternative(s)?

Response: If we calculate the mean (average) ranking for each alternative across all MCDA methods, we get a consensus that can be used as a comprehensive ranking for the alternatives. Based on the given outranking table, the average ranks are: – Alternative 1: Mean Rank = 5.24; Alternative 2: Mean Rank = 6.16; Alternative 3: Mean Rank = 1.26; Alternative 4: Mean Rank = 3.84; Alternative 5: Mean Rank = 2.03; Alternative 6: Mean Rank = 3.57; and Alternative 7: Mean Rank = 6.39. Therefore, the consensus among the methods could be ordered as follows: 1. Third Alternative; 2. Fifth Alternative; 3. Sixth Alternative; 4. Fourth Alternative; 5. First Alternative; 6. Second Alternative; and 7. Seventh Alternative This particular ordering means Alternative 3 is the most preferred, Alternative 5 is the next most preferred, and so on. The seventh alternative is the least preferred based on the ranking by these MCDA methods. [sic]

Prompt 5: Are there any unexpected rankings for certain alternatives when comparing across methods?

Response: Yes, there are some unexpected rankings when comparing across different MCDA methods. Here are few instances: 1. In the second alternative, there is a variation in the ranking of CODAS and MARCOS which placed it at second position whereas all other methods placed it at sixth or seventh position. 2. In the third alternative, there is a significant deviation in VIKOR's ranking, which placed it at sixth position, whereas all other methods placed it at first position. 3. Similarly, in the fourth alternative, the ranking by CODAS and GRA method is deviating from other methods. 4. In the fifth alternative, MAUT has ranked the alternative as fifth, while most other methods rank it as second. 5. The seventh alternative shows great discrepancy. It was ranked seventh by most methods, but ranked fifth by some like GRA and Promethee IV. Such discrepancies and unexpected rankings could be because each MCDA method uses a different set of criteria weights or preference function, and they might approach the problem from different perspectives or assumptions. [sic]

Prompt 6: Are there any methods that consistently rank alternatives differently than most other methods?

Response: From the given outranking table, we can see that some MCDA methods consistently rank alternatives differently than most other methods. For instance: – "codas" consistently ranks the second alternative as the best (second rank), while most other methods rank it anywhere between sixth and seventh. – "cradis" method, alternately, ranks the third alternative as second-best (second rank), whereas the majority of other methods rank it as the best (first rank). – "vikor" method ranks the first alternative as third and the third alternative as sixth, which is quite distinct from other methods that rank those alternatives as sixth and first, respectively. It is also important to mention, not only these methods predict different ranking but also the degree of differences in their results vary. Some methods rank certain options significantly differently than other methods, while some methods show relatively smaller differences. The degree of difference is important information for decision-makers, because it reflects the level of uncertainty or disagreement about the ranking of the alternatives. It might be beneficial to use multiple methods for decision-making to gain comprehensive insights and to consider various perspectives. In addition, it is recommended to understand the underlying principles and assumptions of each MCDA method as they significantly influence the ranking output. Different methods may yield different results due to these operating principles and assumptions, offering different insights into the decision problem. [sic]

Prompt 7: What is the most common ranking for each alternative across all methods?

Response: Based on the table given, below is the most common ranking for each alternative: 1. The most common ranking for the first alternative is 5. 2. The most common ranking for the second alternative is 6. 3. The most common ranking for the third alternative is 1. 4. The most common ranking for the fourth alternative is 3 or 4. 5. The most common ranking for the fifth alternative is 2. 6. The most common ranking for the sixth alternative is 3 or 4. 7. The most common ranking for the seventh alternative is 7. [sic]

Using ChatGPT to analyze the data from Table 4 has added a nuanced layer of understanding that extends beyond what traditional analytical methods offer. The model provided insights into the congruencies and divergences across various MCDA methods, showcasing its value for initial data interpretation and hypothesis formation. Responses to specific prompts like Prompt 4, which discussed a consensus among methods based on mean ranks, require caution since averaging ordinal rankings can be misleading. Prompt 7 shows the most commonly occurring ranks (mode) and is the correct approach to dealing with ordinal data. It is crucial to remember that while ChatGPT provides beneficial initial insights, it doesn't replace the need for rigorous analytical scrutiny. Researchers should use the model as a complement to, not a replacement for, conventional research methods.

4.3 First case study – correlation analysis

To examine the relationships between different methods more deeply, as presented in Table 4, we will use the Kendall Tau correlation coefficient for our analysis. This choice is particularly apt given that Kendall Tau is primarily designed for ordinal data, aligning well

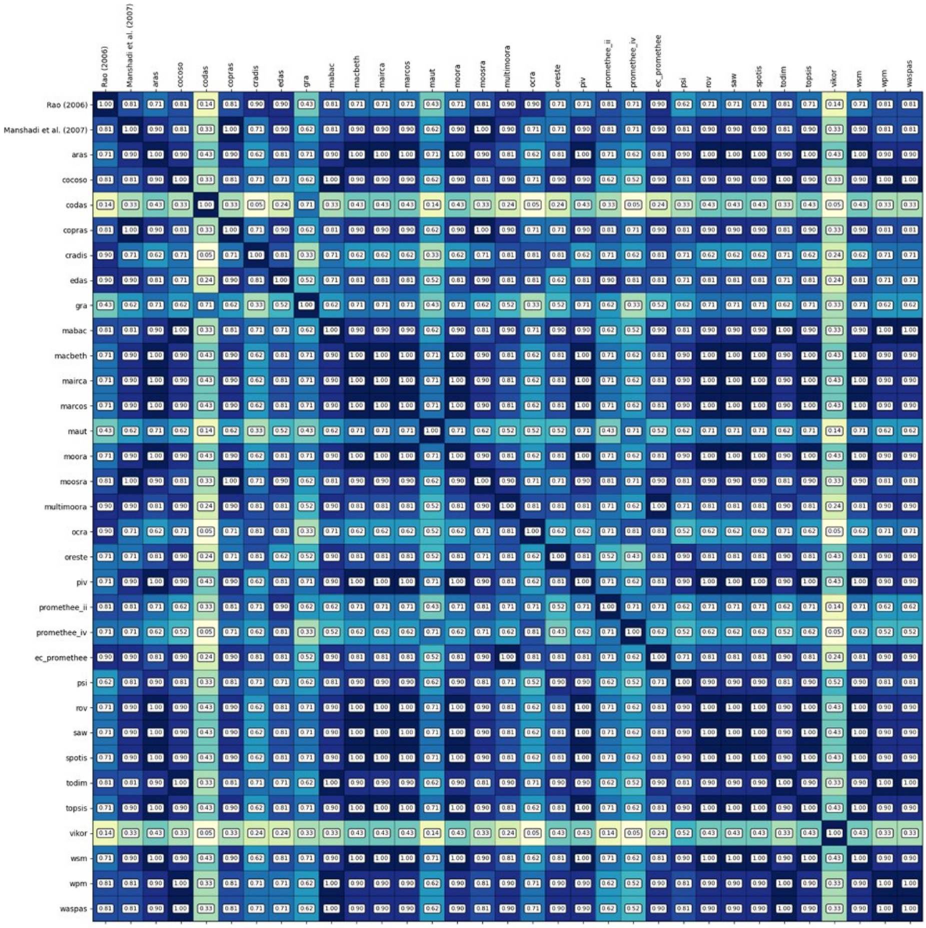

Figure 13. First case study rank correlation
Source: Authors' own creation

with the nature of this problem. Figure 1 offers a color-coded matrix to visualize these correlations. Positive relationships between methods are indicated by varying shades of blue, with darker blue signifying stronger positive correlations. Conversely, shades of green and yellow represent correlations near zero, highlighting the less positive relationships between methods. Notably, the matrix reveals no negative correlations between the methods, underscoring a general agreement trend, albeit to varying degrees.

Figure 13 provides a comprehensive view of the correlations between the various MCDA methods. Excluding the diagonal values, which naturally stand at 1.0, the overall minimum correlation observed was 0.05, while the maximum reached 1.0. Notably, the methods CODAS and VIKOR display the least congruence with other methods and between themselves, both holding a correlation of 0.05, indicating that these methods might offer unique perspectives compared to the others in the matrix.

4.4 First case study – correlation analysis – large language model aid

To extract deeper insights and potentially uncover hidden patterns in Figure 1, we'll leverage the analytical capabilities with ChatGPT (model gpt4). We plan to use a sequence of targeted prompts to augment our initial analysis by shedding light on aspects we might have overlooked.

Prompt 1: Explain the significance of analyzing the correlation of ranks between different MCDA methods.

Response: The purpose of analyzing the correlation of ranks between different MCDA methods is to understand how consistent the methods are with each other. In other words, it establishes the degree to which one method tends to give the same rankings as another method. Looking at the correlation table: – High correlation values (near 1) between any two methods mean that the two methods rank alternatives similarly. For instance, "*Rao* (2006)" and "Manshadi *et al.* (2007)" show a correlation value of 0.809524 indicating a strong positive agreement between these two. – Low correlation values (near 0) indicate that the two methods often rank alternatives very differently. For instance, "Rao (2006)" and "Codas" show a correlation value of 0.142857 indicating low agreement between these two. This analysis helps researchers or decision-makers understand which methods are most consistent with each other, which could be important when choosing which method to use or when interpreting results. If certain methods consistently give similar rankings, a researcher might feel confident using any of them. However, if some methods often disagree with others, it could be an indication that these methods are taking into account some factors that others are not, or weighing some factors more heavily. In this context, understanding these inconsistencies and the reasons for them could yield important insights about the limitations and advantages of different MCDA methodologies. Moreover, if the decision is sensitive or critical, decision-makers may opt to use the most consistent methods or a set of methods that produce similar rankings to ensure the robustness and reliability of the chosen solution. [sic]

Prompt 2: What are the implications if there is a high correlation between the ranks produced by different MCDA methods?

Response: A high correlation between ranks produced by different MCDA methods would suggest a level of agreement or consistency in the decision-making process. This can have several implications: 1. Reveals Robustness: A high correlation among different methods signifies robust results. This means the decision or rank obtained is not sensitive to the specific MCDA method used. 2. Confirms Validity: If multiple methods are showing similar outcomes, it adds to the validity of the results. This can enhance trust and confidence in the outcomes of the decision-making process. 3. Simplifies Process: Knowing that different methods yield very similar results can simplify the decision-making process. Decision-makers might choose to use one method over another based on ease, simplicity or computational efficiency without significantly impacting the result. 4. Indicates Redundancy: However, in case you used various MCDA methods to ensure an extensive and diverse analysis of decision-making factors, high correlation could mean redundancy. If multiple methods provide very similar results, it may not be necessary to implement all of them. 5. Agreement in outcomes: A high correlation indicates good agreement between the rankings. Any selected action or trend will probably be the same or very similar no matter which MCDA method is used. Finally, high correlation between ranks doesn't necessarily guarantee high correlation between the actual numerical scores given, as the methods for arriving at these can be very different depending on the MCDA method used. [sic]

Prompt 3: What might cause a low correlation in rankings between different MCDA methods?

Response: A low correlation between different MCDA methods might result from several factors: 1. Different MCDA methods may take into account different aspects of the alternatives, hence, reflecting divergent perspectives in the decision-making process. 2. The methods analyzed might implement completely different theoretical models, hence, leading to dissimilar conclusions. 3. The divergence can also occur due to the different assumptions that each method makes about the decision problem. 4. In addition, the chosen outputs of the methods might vary greatly. Some methods may output complete rankings, while others might classify decision alternatives into preference groups. 5. The methods may also weight the criteria differently, and small changes in weights can substantially modify the final ranking. Some methods are more sensitive to these weights than others. 6. Finally, the methods might define and interpret uncertainty differently. This can make a huge difference in certain scenarios where the data contains significant uncertainties. [sic]

Prompt 4: What precautions or considerations should be taken when comparing the rankings of different MCDA methods?

Response: When comparing the rankings of different MCDA methods, it is important to bear the following considerations in mind: 1. Not all MCDA methods evaluate criteria in the same way. Some methods may give total priority to the best performing alternative, while others play with compensations. 2. An MCDA method may not work in the same way for all kinds of problems. For instance, some MCDAs require a linear preference function, meaning that they are not suited to nonlinear problems. 3. Different MCDA methods may involve different assumptions and logic. For instance, some may assume compensation between criteria, whereas others do not, etc. 4. The results of each MCDA method can markedly vary depending on how the analysis was set up. For instance, the weight assigned to different criteria can have a major impact on results. 5. It is important to take into account that correlation values are not precise indicators of similarity between rankings produced by different methods. Given such considerations, rather than comparing the end ranking of different MCDA methods directly, it may be more informative to examine the process that each method uses to arrive at its ranking. This may give you a better appreciation of why different methods produced different results and more confidence in the method that best matches your intuition and understanding of the problem at hand. [sic]

Using ChatGPT with a researcher's expertise can offer fresh perspectives on a given issue. However, it is vital to recognize its limitations in capturing every nuance of a broad topic. Specific and directed prompts tend to yield more targeted responses. In contrast, open-ended inquiries might miss certain critical aspects, as observed in the case of the discrepancies between VIKOR and CODAS compared to other methods. Despite the insights provided by ChatGPT, human oversight remains indispensable, especially when discerning evident patterns like those in our example.

5. PyDecision – large language model – second case study

Again, this section aims to demonstrate the versatility of pyDecision in conjunction with ChatGPT. The second case study explores various methods capable of generating weights, including two methods from the academic literature not natively supported by pyDecision. Then, ChatGPT plays an integral role in interpreting the outcomes. We use a sequence of targeted prompts to guide the decision-maker in evaluating and understanding the results. In this section, we showcase the combination of pyDecision and ChatGPT. Our second case study examines diverse weight-generating methods, also featuring two techniques from academic literature not inherently integrated into pyDecision. Again, ChatGPT is used through a series of prompts, and we try to comprehend the results better.

The second case study can be accessed at < https://bit.ly/3QAaJle >.

5.1 Second case study – methods comparison

The second case study is based on the research conducted by Bottero *et al.* (2015), where they developed a framework for multicriteria decision-making using the ELECTRE III method. The study used this framework based on six criteria to thoroughly assess five projects labeled a1 to a5. Investment Costs (IVC) is the first criterion focusing on construction expenditures. The criterion of profitability (PRF) is followed by evaluating not only the direct financial gains but also the broader economic benefits of a project using a qualitative scale. Likewise, New Services for the Population (NSP) measures a project's contributions to communal amenities such as recreational spaces or green areas. Landscape Ecology (LSE) examines a project's ecological footprint, precisely its effect on local biodiversity, quantified in hectares of naturalized land. Environmental Effects (EVE) scrutinizes the potential impacts on physical and ecological aspects, like hydrology and geotechnical conditions, and uses a qualitative scale. Finally, the binary criterion of Consistency with Local Planning Requirements (CPR) ascertains if a project adheres to existing regulatory frameworks. All the criteria are designed for maximization except for Investment Costs (IVC), a minimization criterion.

Table 5 shows the values that represent the performance of each project (a1–a5) against these criteria. For instance, project a1 has an investment cost of 30,000 and scores 3 in profitability, while project a5 has an investment cost of 900,000 but ranks highest in environmental effects with a score of 7. Table 6 delineates the weights assigned to the criteria based on two distinct methods: Bottero *et al.* (2015) and Rodrigues *et al.* (2021). For each criterion, weights signify its importance in the overall decision-making process. Rodrigues *et al.* (2021) used an algorithm named Ranking Trees to calculate the weights for each criterion. In contrast, Bottero *et al.* (2015) relied on domain specialists to determine the weightage.

Table 7 shows the MCDA methods BWM, CILOS, CRITIC, Entropy, IDOCRIW and MEREC; additionally, two custom methods have been integrated, Bottero *et al.* (2015) and Rodrigues *et al.* (2021). For BWM, there's an emphasis on identifying the Most Important

Table 5. Second case study

Alternatives	IVCI (MIN)	PRF (MAX)	NSP (MAX)	LSE (MAX)	EVE (MAX)	CPR (MAX)
a1	30,000	3	1	2	4	1
a2	45,000	3	5	5	5	1
a3	90,000	1	6	3.2	7	1
a4	120,000	1	7	3.5	6	1
a5	900,000	7	7	1	3	0

Source(s): Authors' own creation

Table 6. Second case study – weights

Criteria	Bottero *et al.* (2015)	Rodrigues *et al.* (2021)
IVCI	0.140	0.170
PRF	0.250	0.130
NSP	0.080	0.100
LSE	0.200	0.230
EVE	0.310	0.190
CPR	0.020	0.180

Source(s): Authors' own creation

Table 7. Second case study – results

Methods	IVCI	PRF	NSP	LSE	EVE	CPR
Bottero *et al.* (2015)	0.140	0.250	0.080	0.200	0.310	0.020
Rodrigues *et al.* (2021)	0.170	0.130	0.100	0.230	0.190	0.180
BWM	0.178	0.119	0.084	0.124	0.416	0.078
CILOS	0.050	0.059	0.361	0.08	0.116	0.327
CRITIC	0.140	0.313	0.178	0.106	0.114	0.148
Entropy	0.288	0.236	0.116	0.109	0.039	0.212
IDOCRIW	0.361	0.166	0.117	0.114	0.086	0.156
MEREC	0.065	0.055	0.074	0.028	0.014	0.763

Source(s): Authors' own creation

Criteria (MIC = 2, 4, 5, 3, 1 and 6) and Least Important Criteria (LIC = 6, 1, 3, 5, 4 and 2), represented as rankings from 1 to 6 for each criterion in each situation; therefore, the best criterion is EVE (first in MIC) and the worst criteria PRF(first in LIC).

The second case study shows that BWM has a unique weight distribution, emphasizing the EVE criterion. CILOS seems to emphasize NSP and CPR more, while CRITIC prioritizes PRF. Entropy allocates the highest weight to IVCI, while IDOCRIW does the same but with an even more pronounced emphasis. Finally, MEREC stands out by assigning most of its weight to CPR.

5.2 Second case study – methods comparison – large language model aid

To extract deeper insights and potentially uncover hidden patterns in Table 7, we'll leverage the analytical capabilities with ChatGPT (model gpt4). We plan to use a sequence of targeted prompts to augment our initial analysis by shedding light on aspects we might have overlooked.

Prompt 1: Which methods are more similar and which ones are more dissimilar?

Response: To determine the similarity and dissimilarity between different MCDA methods, various statistical analysis methods can be used. Among those, correlation analysis and clustering analysis are the most commonly used. For this specific type of data (weight of each criterion calculated by each MCDA method), correlation analysis would be a good choice. This analysis measures the statistical relationship, but not deterministic, between two sets of variables. High positive correlation value (close to 1) means two methods are very similar – they tend to give higher weights to the same criteria and lower weights to the same set of other criteria. On the other hand, high negative correlation value (close to -1) means two methods are very dissimilar – they tend to give a higher weight to a criterion that the other method gives lower weight, and vice versa. However, from the table alone, we can't definitively state which methods are most similar or dissimilar as it would require computation and analysis. As per the rough estimate based on the provided weights, it looks like "Bottero *et al.* (2015)*" and* "Rodrigues *et al.* (2021)" are somewhat similar. The "cilos" method seems to be relatively dissimilar comparing with others, based on its distinct patterns of weights assignment. But please note that these are only heuristic observations. For an accurate similarity/dissimilarity evaluation, quantifiable statistical measures such as correlation coefficients should be computed. [sic]

Prompt 2: Are there certain criteria that consistently receive high weights across all methods? What might these key criteria suggest about the decision problem at hand?

Response: The information from the table seems to be a bit incomplete for providing in-depth explanations. However, based on what's given, it seems that there might be certain criteria that receive high weights across different methods when averaged. For example, first and fifth criteria average to relatively high weights across all methods, suggesting they might be particularly important for the decision problem at hand. In MCDA, the weights assigned to different criteria reflect their relative importance in the decision problem. Hence, these weights suggest that these two factors/criteria may be of considerable significance, potentially impacting the overall decision to a great extent. However, the specifics about what these criteria represent or the decision problem at hand are not given in the question, which limits the analysis depth. That said, keep in mind that even though a criterion might receive high weights on average, it doesn't mean it is always the most significant– different methods may prioritize different aspects, reflecting varying perspectives on the same decision problem. In addition, the decision-making context (whether you're selecting the best alternative, ranking alternatives, sorting them into categories, etc.) also affects the interpretation of such weights. The comparison of the weights across different methods might be used to validate the robustness of the results (if similar weights are obtained, confidence in these can be increased) or to explore how sensitive the results are to the specific weighting method used. Note that the high weight in the last column (0.763368) is potentially an outlier or it might be indicating a very significant criterion for merec method. Without contextual information, it is impossible to provide definite explanations. [sic]

Prompt 3: Conversely, are there criteria that consistently receive low weights across all methods? This could indicate aspects that are less important to the decision context.

Response: It appears that the later methods (cilos, critic, entropy, idocriw, merec) consistently attribute lower weights to the fifth criterion (row 5), with merec attributing the lowest weight of all (0.013805). This suggests that the fifth criterion may be of less importance in these MCDA methods. Interestingly, the earlier methods (Bottero *et al.*, 2015*;* Rodrigues *et al.*, 2021, bwm) attribute more importance to the fifth criterion. This could indicate a difference in prioritization between different methods. Also, the sixth criterion seems to receive a very low weight from the methods Bottero *et al.* (2015) and Rodrigues *et al.* (2021), which might mean that these methods don't consider the sixth criterion to be very important. However, further analysis would be needed to determine whether these patterns hold true across different decision contexts and to interpret what this means in each specific context. [sic]

Prompt 4: How much variability is there in weights assigned to each criterion by different methods? High variability could suggest that different methods interpret the importance of the criteria differently.

Response: To determine the variability in weights assigned to each criterion by different methods, it would be useful to calculate the standard deviation for each row. The standard deviation will show how much variation exists from the average (mean) weight. For example, let's calculate the standard deviation of the weights assigned to the first criterion by each method: – Value by Bottero *et al.* (2015): 0.14; Value by Rodrigues *et al.* (2021): 0.17; Value by bwm: 0.178310; Value by cilos: 0.039830; Value by critic: 0.140939; Value by entropy: 0.288163; Value by idocriw: 0.360795; Value by merec: 0.065421. We could use a standard deviation calculator or a software like Excel for this calculation. Then, we'd repeat this process for each row (i.e. each criterion) to see how much variability exists in the importance assigned to each criterion by the different methods. After obtaining all the standard deviations, if they're high, then it suggests that the methods interpret the importance of the criteria differently. But if they're low, then it suggests that the methods interpret the importance of the criteria similarly. Note that I did not provide the actual standard deviation calculations in these examples because this assistant doesn't have access to computational functions for calculating this. [sic]

Prompt 5: Are there any noticeable correlations between the weights assigned by different methods

Response: Based on the data provided, it is hard to determine any solid correlations just by visual assessment. However, few patterns can be observed: 1. Bottero *et al.* (2015) and Rodrigues *et al.* (2021): the weightings for the third, fourth and fifth criteria from these two methods tend to move in the same direction, i.e. when one increases the other increases as well, and when one decreases the other also decreases. 2. Bottero *et al.* (2015) and Cilos: for the first and third criterias, when Bottero *et al.* (2015) has a low rating, Cilos displays a high rating, suggesting an inverse relationship. 3. Rodrigues *et al.* (2021) and IDOCRIW: for the first and second criterias, when Rodrigues *et al.* (2021) moves in one direction, IDOCRIW moves in opposite directions. To conclude on any correlations and their significance, a statistical analysis (Pearson correlation, Spearman correlation, etc.) would be needed to analyze the numerical patterns within and between the different MCDA methods. This includes calculating correlation coefficients which could lend a quantitative understanding to any relationships between these methods. AI models would not be able to conclusively identify correlations through purely visual inspection of the data. [sic]

Prompt 6: Can you identify outlier methods that assign weights significantly different from others?

Response: To identify the outlier MCDA methods, one way is to compare each method's weights to the median (or mean) of all the methods for each criterion. Let's consider outlier methods as those which deviate from the others by more than a certain threshold, say 0.2. Bottero *et al.* (2015) and Rodrigues *et al.* (2021) seem to assign similar weights to each criterion. Most of their weights fall in the same range. For "bwm," weights are more variable. The weight of the fifth criterion is the highest among all the methods, which could make "bwm" an outlier for this specific criterion. "cilos" seems to assign significantly different weights to the third and sixth criteria, which are way higher than the other methods. And so, it can be considered an outlier for these criteria. "critic" falls in the same pattern as "bwm" with higher weights for the second criterion. "entropy" assigns the most weight to the first criterion which is higher than other methods. "idocriw" has relatively low variability in weights across the criteria but still the first criterion has the highest weight which is a bit higher than others. "merec" features a very high weight for the last criterion, which could make it an outlier method for this criterion. It is important to note that the relevance of these differences depends on the domain of application, and on the specific method's inherent characteristics. There may be good reasons for these differences, and they may or may not indicate actual problems. [sic]

Prompt 7: Is there a specific method that consistently assigns higher or lower weights to all criteria? If so, what does this indicate about the method's evaluation approach?

Response: Based on the table given, it is hard to pin down a method that consistently assigns higher or lower weights to all criteria without knowing the specific criteria being evaluated. However, conceptually, if a method assigns consistently higher weights to all criteria, it may handsomely value all aspects being measured, making it a broad-brush approach that might be good for general analyses but lack specificity. In contrast, if a method consistently assigns lower weights to all criteria, it may indicate that the method is more conservative or restrictive in its weighting, with a more skeptical approach in acknowledging the impact of each criterion. Observations from the table do indicate that, "idocriw" can be seen to assign relatively higher weights overall, while "cilos" and "merec" generally assign lower weights. This could mean "idocriw" gives generous importance to all criteria, viewing a wide variety as potentially impactful. "cilos" and "merec" methods could be more conservative or selective in their approach, or they might assign higher weights to criteria not listed in this table. It is definitely crucial, though, to possess knowledge of the specific criteria and understanding the algorithms behind each MCDA method to make accurate insights.

Prompt 8: Do the weightings across different methods suggest a consensus on the importance ranking of the criteria?

Response: The results are quite diverse across different MCDA methods, indicating a lack of consensus on the importance ranking of the criteria. Some methods give more importance to the first criterion (Bottero *et al.*, Rodrigues *et al.*, bwm, critic, idocriw), but others give more weight to the second (entropy, merec) or even the sixth criterion (cilos). So, the importance of the criteria largely depends on the MCDA method applied. It would be helpful to further investigate the underlying reasons and assume a weighting method according to the specific nature and requirements of the decision problem.

ChatGPT, when operating without context, might sometimes plunge into foundational explanations of concepts that researchers might already be familiar with. For instance, as seen in Prompts 4 and 5, the responses can occasionally be rudimentary or lacking depth. Providing specific context or applying domain expertise is crucial to derive the most valuable and insightful answers.

5.3 Second case study – correlation analysis

We will use the Pearson correlation coefficient as our analytical tool to examine the interconnections between continuous values of the MCDA methods showcased in Table 7. Figure 2 provides a color-coded matrix of these interrelationships. Shades of blue, growing deeper with intensity, denote positive correlations between methods. In contrast, hues of green and yellow symbolize near-zero and negative correlations, respectively. Remarkably, the matrix underscores pronounced negative correlations existing between certain methods.

Upon examining the results from Figure 14, several standout relationships can be identified. Most prominently, CILOS and MEREC share a solid positive correlation of $r = 0.85$, suggesting that these methods often move in tandem. ENTROPY and IDOCRIW also display a strong positive correlation of $r = 0.83$, indicating a similar trend. On the other end of the spectrum, the results of Bottero *et al.* (2015) and CILOS exhibit a pronounced negative correlation of $r = -0.76$. Another significant negative relationship is observed between CRITIC and the results from Rodrigues *et al.* (2021), with a correlation coefficient of $r = -0.66$.

5.4 Second case study – correlation analysis – large language model aid

To extract deeper insights and potentially uncover hidden patterns in Figure 2, we'll leverage the analytical capabilities with ChatGPT (model gpt4). We plan to use a sequence of targeted prompts to augment our initial analysis by shedding light on aspects we might have overlooked.

Prompt 1: Explain the significance of analyzing the correlation of weights between different MCDA methods.

Response: Analyzing the correlation of weights between different MCDA methods is significant in several ways: 1. Checks consistency: by examining the correlation, we can check the consistency of the decision-making process. If different MCDA methods are giving conflicting results for certain weight comparisons, it indicates potential inconsistency and warrant a re-examination of the decision problem's structure and data. 2. Validating MCDA method: this correlation data can also help in validating and comparing the reliability of different MCDA methods. If two methods show a high correlation of weights, they are likely to be similarly effective. In contrast, a low correlation might indicate one method is not as reliable. 3. Uncovers biases: the correlation analysis can also uncover potential biases that

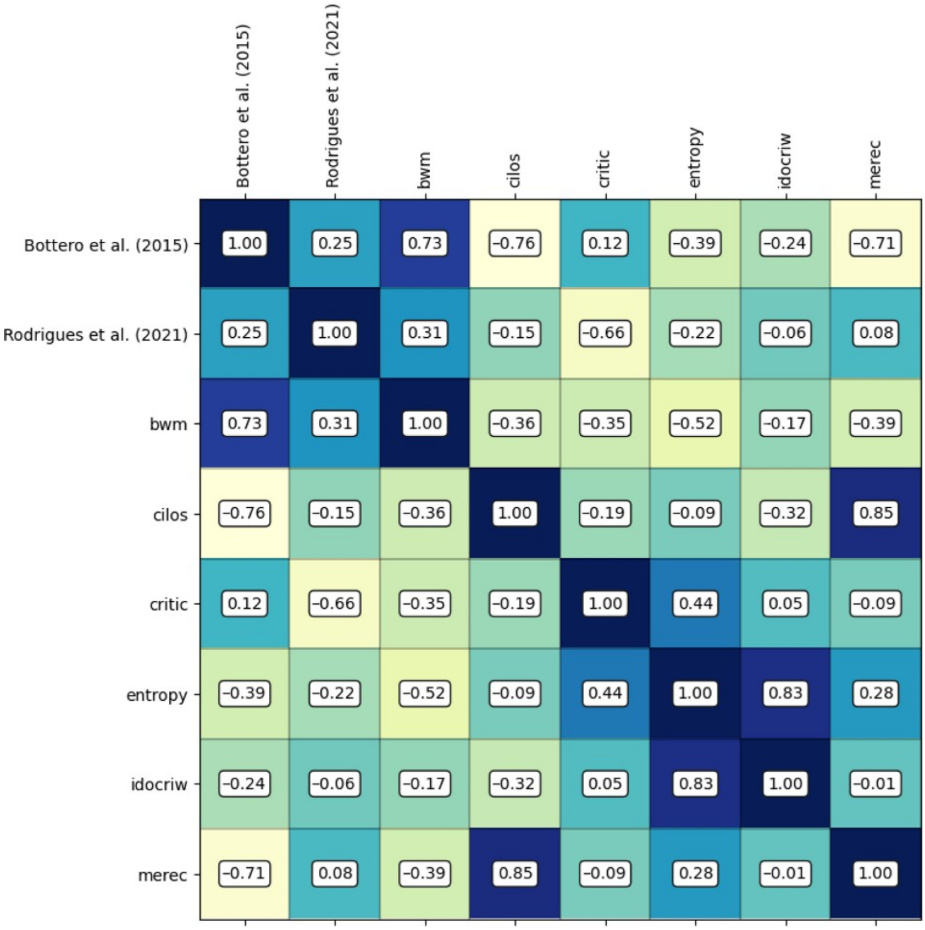

Figure 14. Second case study weights correlation
Source: Authors' own creation

might not be evident when each method is considered independently. 4. Improving decision-making process: high correlation strengthens our confidence in the decision-making process, while low correlation could stimulate further investigation to uncover the potential reasons and subsequently improve the decision-making process. 5. Integrating multiple MCDA methods: if the weights' correlation between two methods is found to be high, one might think of integrating the two methods into a "hybrid" approach which may potentially improve the robustness of the model. In the given table, the positive values suggest a positive correlation, i.e. as the weight increases in one method, it also increases in the other. On the other hand, negative values suggest a negative correlation, i.e. as the weight increases in one method, it decreases in the other method. The values close to 1 or −1 indicate strong positive or negative correlation, respectively, while those near 0 indicate weak or no correlation. [sic]

Prompt 2: What might cause differences in the weighting of criteria across various MCDA methods?

Response: The differences in the weighting of criteria between various MCDA methods can be caused by several factors: 1. Different interpretations of criteria importance: the perceived importance of a criterion can change based on who is assessing the criteria. Different people may prioritize different elements based on their perspectives. For instance, experts and laypersons, or different demographic groups, will likely perceive the importance of individual criteria differently. 2. Choosing different methods: different MCDA methods use different algorithms that interpret the importance of a criterion differently. They use distinct efficacy measures. For example, the AHP uses a pairwise comparison approach to defining weights, while the PROMETHEE method uses a preference function. 3. Variations in the form and type of data: different MCDA methods are better suited to different types of data (quantitative, qualitative, interval, ordinal, etc.), which can affect the weighting. 4. Sensitivity to uncertainty: some MCDA methods are more sensitive to uncertainty than others. If the data or preferences are uncertain, the methods may give different weights to different criteria. 5. Context and domain of application: the specific application or context of the decision problem might influence the weighting process. For example, in environmental decision-making, ecological criteria could be weighed more heavily than economic factors, and this could shift depending on the method. 6. Bias and subjectivity: as MCDA reflects human judgment, personal biases and perceptions could influence the choice of criteria and the assigned weights. These can vary depending on the MCDA method being used. [sic]

Prompt 3: How can the correlation of weights between different MCDA methods impact the final decision?

Response: The correlation of weights between different MCDA methods can greatly impact the final decision as it determines the agreement or disagreement between these methods. 1. Positive correlation: if two MCDA methods have a high positive correlation like the Bottero *et al.* (2015) method and bwm method, this means that they broadly agree in the way they assign weights to the criteria. A decision made using these methods would likely be similar as they deem the same criteria to be important. 2. Negative correlation: if two MCDA methods have a high negative correlation like the *Bottero et al.* (2015) method and the cilos method, this means they disagree in how they rank the importance of the criteria. A decision made using these methods could be very different which could complicate the decision-making process. 3. No correlation: if two MCDA methods have correlation near zero (For instance, eigenvalue and idocriw method), this means there is no linear association between the weights assigned by these methods. Without additional information, we cannot deduce how a decision would vary if different combinations of these methods were used. Highly correlated MCDA methods may help to reinforce a decision if they converge towards the same outcome, thus, providing additional confidence in the decision made. However, if there is disagreement or lack of correlation between MCDA methods, this could suggest that the decision is more complex, with different methods highlighting different priorities. Therefore, the decision-makers may wish to scrutinize the differences, investigate the underlying causes and ultimately, choose an approach that best suits their problem and context. Remember that correlation does not imply causation – while it shows the relationship, it does not provide insights into why these weights are influencing each other in such way. It is also crucial to consider, MCDA is only one step in decision-making process, and decisions should not be made solely on the results of MCDA. Other factors, including expert opinion, regulatory considerations and public response, also need to be taken into account. [sic]

Prompt 4: What could be the implications if there is a high correlation of weights across different MCDA methods?

Response: If there is a high correlation of weights across different MCDA methods, it means that the methods are giving almost similar and identical results. This could have several implications: 1. Consistency: in a decision-making process, consistency is key, and, thus, a high correlation could signify that there is a high level of consistency in the decision-making process. 2. Confirmation of validity: strong correlation may serve as validation that the methods applied are giving accurate and valid outcomes. If similar weights are attributed to the same alternatives across different MCDA methods, it is more likely that the results are a true reflection of the situation. 3. Ease of decision-making: if there is a high correlation across different MCDA methods, it means that there is agreement on which alternatives are preferred, making it easier to make a final decision. However, it is important to keep in mind: 1. Risk of redundancy: if all methods are giving almost similar results, using different MCDA methods may be inefficient, as effort and resources may be spent on applying and analyzing multiple methods when one might have been sufficient. 2. Limits innovation: there may be instances where a decision-maker is looking for diversified analysis and results. In such cases, a high correlation of weights reduces the scope for finding innovative solutions. 3. Reliability: a high correlation might also suggest that the methods are not as reliable as they seem, as they do not provide different perspectives on the decision problem. It could indicate that the methods are being influenced by the same underlying factors rather than each providing a unique analysis. Overall, understanding these correlations can create opportunities to use the best suited MCDA method for a given context or to blend results from various methods for a more comprehensive analysis. [sic]

Prompt 5: What strategies can be used to address inconsistencies in the weights assigned by different MCDA methods?

Response: Looking at the table above, it is apparent that there are some notable differences and inconsistencies in the weights assigned by different MCDA methods. Here are some strategies that could be used to address these inconsistencies: 1. Decision matrix: use a decision matrix to compare the weightings assigned by different methods. This may involve ranking the methods based on their ability to accurately reflect decision-makers' preferences. 2. Comparison of methods: all the MCDA methods may not be equally valid for the kind of data or decision issue in hand. Carry out a comparison of the methods to understand which is the most appropriate for the specific decision context, and prioritize their weightings/refine them accordingly. 3. Average: you may consider averaging the weights determined by all methods for each criterion. This simple method is useful to address inconsistencies between different methods, although it may not always lead to the most desirable outcomes. 4. Sensitivity analysis: conduct a sensitivity analysis to study the effects of changes in the input data (i.e. changes in the weights assigned by different MCDA methods) on the final decision. 5. Meta-modeling: meta-models are a type of model that summarizes the knowledge gained from running an original model several times. This can be used for reconciliation of the inconsistencies. 6. Expert judgement: incorporate the judgment of the decision-makers or experts in refining the weights. It may also be helpful to consider qualitative views on the relative importance of different criteria. 7. Weighting techniques: use well-established weighting techniques that could better encapsulate the decision-maker's preferences. Techniques such as the AHP or the BWM could be used. 8. Cross-check: cross-check with real-world outcomes to see if one method's weights tend to be more accurate than the others. Remember, these strategies are not mutually exclusive and can often be used in conjunction with one another to arrive at the best possible outcome. [sic]

ChatGPT can occasionally misinterpret the nuances of specific techniques or methods. For example, Prompt 2 correctly described AHP's approach to weight generation but inaccurately claimed that the PROMETHEE method uses a preference function for the same purpose. While the model can offer valuable insights, it is essential to cross-check its responses for accuracy.

6. Large data sets

To assess the computational performance of various MCDA methods under increasing levels of complexity, we will evaluate their execution times across a range of matrix sizes, representing alternatives and criteria. These matrix sizes include 10 × 10, 10 × 25, 10 × 50, 10 × 100, 25 × 10, 25 × 25, 25 × 50, 25 × 100, 100 × 10, 100 × 25, 100 × 50 and 100 × 100. We will perform 100 independent runs for each configuration, recording both the average execution time (in seconds) and the standard deviation to ensure reliable and statistically robust results. Certain methods that assign weights to criteria are unsuitable for larger-scale stress testing because of the meaningfulness and quality of the results. This limitation arises primarily because, as the number of criteria increases, the differences in the calculated weights often diminish, leading to nearly uniform weights across all criteria. Similarly, methods like AHP, specifically designed for smaller, well-structured problems, lack validation for significantly large data sets and are rarely used in such contexts. Consequently, the methods selected for stress testing are those listed in Table 4, as they are more appropriate for addressing larger and more complex decision-making scenarios.

Table 8 reveals clear distinctions in terms of computational efficiency. A group of methods, COPRAS, SPOTIS, MOORA, MACBETH, SAW, CRADIS, MARCOS and TOPSIS, executes with remarkably low overhead (often under 0.001 seconds) across small-to-medium problems (e.g. 10 × 25 and 25 × 50). Their runtime remains minimal even as problem dimensions increase to 100 × 100, seldom exceeding a few milliseconds on average. This group's near-linear or even sublinear scaling suggests straightforward mathematical formulations and efficient implementations, making them well-suited for scenarios involving relatively large alternative–criteria matrices.

A second group of methods, which includes EDAS, MAIRCA, MULTIMOORA, MABAC and MAUT, exhibits moderate yet still quite feasible runtimes, generally in the low millisecond range for moderate scales, occasionally rising toward tens or hundreds of milliseconds at 100 × 100. Their higher complexity stems from more elaborate transformations or partial iterative steps within the decision-making workflow. Nonetheless, the standard deviations of their runtimes remain modest, reflecting stable performance characteristics. Such methods can be comfortably applied in practical contexts with up to a few hundred alternatives and criteria, as even repeated analyses or sensitivity checks would not pose a prohibitive computational burden.

By contrast, the PROMETHEE family (particularly PROMETHEE II, PROMETHEE IV and EC PROMETHEE) and TODIM demonstrate markedly higher execution times when the matrix size grows, with EC PROMETHEE reaching around 6 s at 100 × 100. Despite the comparatively longer processing, standard deviations remain small in most cases, signaling consistency in computational load. We can note that methods characterized by simple ratio or additive formulations dominate the lower spectrum of execution times. In contrast, iterative or preference-flow-based approaches necessitate increased caution at large scales.

7. Conclusion

In conclusion, the PyDecision library, implementing over 70 MCDA methods, offers a comprehensive toolkit for researchers and practitioners to explore, analyze and solve complex decision-making problems. Its integration into the Python ecosystem provides several advantages over traditional MCDA software. Its open-source nature invites collaboration and continuous improvement, ensuring that the library remains up-to-date with

Table 8. Time results

Matrix size	10 ×10	10 × 25	10 × 50	10 × 100	25 × 10	25 × 25	25 × 50	25 × 100
ARAS	0.0002 ± 0.0000	0.0003 ± 0.0000	0.0006 ± 0.0000	0.0012 ± 0.0000	0.0002 ± 0.0000	0.0004 ± 0.0000	0.0020 ± 0.0001	0.0016 ± 0.0001
CoCoSo	0.0002 ± 0.0000	0.0003 ± 0.0000	0.0006 ± 0.0000	0.0011 ± 0.0001	0.0002 ± 0.0000	0.0003 ± 0.0000	0.0011 ± 0.0002	0.0012 ± 0.0000
CODAS	0.0002 ± 0.0000	0.0003 ± 0.0000	0.0006 ± 0.0000	0.0009 ± 0.0000	0.0007 ± 0.0001	0.0009 ± 0.0000	0.0030 ± 0.0001	0.0019 ± 0.0000
COPRAS	0.0000 ± 0.0000	0.0001 ± 0.0000	0.0001 ± 0.0000	0.0001 ± 0.0000	0.0001 ± 0.0000	0.0001 ± 0.0000	0.0001 ± 0.0000	0.0001 ± 0.0000
CRADIS	0.0001 ± 0.0000	0.0002 ± 0.0000	0.0003 ± 0.0000	0.0006 ± 0.0000	0.0001 ± 0.0000	0.0002 ± 0.0000	0.0008 ± 0.0001	0.0007 ± 0.0001
EDAS	0.0002 ± 0.0000	0.0005 ± 0.0001	0.0008 ± 0.0000	0.0016 ± 0.0001	0.0004 ± 0.0000	0.0011 ± 0.0000	0.0035 ± 0.0002	0.0039 ± 0.0001
GRA	0.0002 ± 0.0000	0.0004 ± 0.0000	0.0008 ± 0.0000	0.0015 ± 0.0001	0.0001 ± 0.0000	0.0004 ± 0.0000	0.0016 ± 0.0001	0.0015 ± 0.0001
MABAC	0.0014 ± 0.0001	0.0032 ± 0.0001	0.0062 ± 0.0001	0.0133 ± 0.0002	0.0039 ± 0.0000	0.0098 ± 0.0001	0.0368 ± 0.0012	0.0398 ± 0.0003
MACBETH	0.0001 ± 0.0000	0.0002 ± 0.0000	0.0004 ± 0.0000	0.0007 ± 0.0000	0.0001 ± 0.0000	0.0002 ± 0.0000	0.0007 ± 0.0003	0.0008 ± 0.0000
MAIRCA	0.0003 ± 0.0000	0.0005 ± 0.0001	0.0011 ± 0.0000	0.0022 ± 0.0000	0.0003 ± 0.0000	0.0006 ± 0.0000	0.0026 ± 0.0005	0.0022 ± 0.0000
MARCOS	0.0001 ± 0.0000	0.0002 ± 0.0000	0.0005 ± 0.0001	0.0010 ± 0.0000	0.0001 ± 0.0000	0.0003 ± 0.0000	0.0011 ± 0.0003	0.0010 ± 0.0000
MAUT	0.0004 ± 0.0000	0.0010 ± 0.0000	0.0020 ± 0.0001	0.0043 ± 0.0001	0.0006 ± 0.0000	0.0014 ± 0.0001	0.0058 ± 0.0007	0.0052 ± 0.0001
MOORA	0.0001 ± 0.0000	0.0001 ± 0.0000	0.0002 ± 0.0000	0.0005 ± 0.0001	0.0001 ± 0.0000	0.0001 ± 0.0000	0.0007 ± 0.0002	0.0005 ± 0.0001
MOOSRA	0.0001 ± 0.0000	0.0002 ± 0.0000	0.0002 ± 0.0000	0.0005 ± 0.0001	0.0001 ± 0.0000	0.0001 ± 0.0000	0.0007 ± 0.0002	0.0005 ± 0.0001
MULTIMOORA	0.0002 ± 0.0000	0.0004 ± 0.0000	0.0006 ± 0.0000	0.0010 ± 0.0000	0.0002 ± 0.0000	0.0006 ± 0.0000	0.0025 ± 0.0004	0.0016 ± 0.0000
OCRA	0.0001 ± 0.0000	0.0004 ± 0.0000	0.0007 ± 0.0001	0.0014 ± 0.0001	0.0002 ± 0.0000	0.0004 ± 0.0000	0.0013 ± 0.0003	0.0017 ± 0.0001
ORESTE	0.0009 ± 0.0000	0.0019 ± 0.0000	0.0037 ± 0.0001	0.0080 ± 0.0001	0.0011 ± 0.0000	0.0024 ± 0.0001	0.0107 ± 0.0007	0.0100 ± 0.0001
PIV	0.0002 ± 0.0000	0.0003 ± 0.0000	0.0007 ± 0.0001	0.0014 ± 0.0000	0.0001 ± 0.0000	0.0003 ± 0.0000	0.0018 ± 0.0003	0.0013 ± 0.0000
PROMETHEE II	0.0013 ± 0.0001	0.0029 ± 0.0000	0.0055 ± 0.0001	0.0119 ± 0.0002	0.0070 ± 0.0001	0.0165 ± 0.0002	0.0703 ± 0.0017	0.0691 ± 0.0007
PROMETHEE IV	0.0041 ± 0.0001	0.0099 ± 0.0002	0.0185 ± 0.0003	0.0366 ± 0.0005	0.0238 ± 0.0003	0.0613 ± 0.0004	0.1767 ± 0.0052	0.2439 ± 0.0022
EC PROMETHEE	0.0030 ± 0.0006	0.0073 ± 0.0006	0.0143 ± 0.0006	0.0271 ± 0.0007	0.0147 ± 0.0005	0.0363 ± 0.0008	0.0802 ± 0.0011	0.1458 ± 0.0012
PSI	0.0001 ± 0.0000	0.0002 ± 0.0000	0.0003 ± 0.0000	0.0006 ± 0.0000	0.0001 ± 0.0000	0.0002 ± 0.0000	0.0008 ± 0.0002	0.0006 ± 0.0000
ROV	0.0002 ± 0.0000	0.0005 ± 0.0000	0.0011 ± 0.0000	0.0021 ± 0.0000	0.0002 ± 0.0000	0.0005 ± 0.0000	0.0023 ± 0.0002	0.0022 ± 0.0000
SAW	0.0001 ± 0.0000	0.0002 ± 0.0000	0.0004 ± 0.0001	0.0007 ± 0.0000	0.0001 ± 0.0000	0.0002 ± 0.0000	0.0008 ± 0.0001	0.0007 ± 0.0000
SPOTIS	0.0000 ± 0.0000	0.0000 ± 0.0000	0.0000 ± 0.0000	0.0000 ± 0.0000	0.0000 ± 0.0000	0.0000 ± 0.0000	0.0002 ± 0.0002	0.0001 ± 0.0000
TODIM	0.0041 ± 0.0000	0.0099 ± 0.0001	0.0199 ± 0.0002	0.0396 ± 0.0003	0.0263 ± 0.0002	0.0722 ± 0.0020	0.1982 ± 0.0069	0.2574 ± 0.0009
TOPSIS	0.0001 ± 0.0000	0.0002 ± 0.0000	0.0004 ± 0.0000	0.0008 ± 0.0000	0.0001 ± 0.0000	0.0004 ± 0.0000	0.0004 ± 0.0000	0.0008 ± 0.0000
VIKOR	0.0002 ± 0.0000	0.0004 ± 0.0001	0.0005 ± 0.0000	0.0009 ± 0.0000	0.0002 ± 0.0000	0.0007 ± 0.0001	0.0005 ± 0.0000	0.0009 ± 0.0000
WSM	0.0002 ± 0.0000	0.0004 ± 0.0000	0.0009 ± 0.0001	0.0016 ± 0.0001	0.0002 ± 0.0000	0.0006 ± 0.0001	0.0009 ± 0.0000	0.0016 ± 0.0001
WPM	0.0002 ± 0.0000	0.0004 ± 0.0000	0.0009 ± 0.0001	0.0016 ± 0.0001	0.0002 ± 0.0000	0.0006 ± 0.0001	0.0009 ± 0.0000	0.0016 ± 0.0001
WASPAS	0.0002 ± 0.0000	0.0004 ± 0.0000	0.0009 ± 0.0001	0.0016 ± 0.0001	0.0002 ± 0.0000	0.0006 ± 0.0001	0.0009 ± 0.0000	0.0016 ± 0.0001

(*continued*)

Table 8. Continued

Matrix size	50 × 10	50 × 25	50 × 50	50 × 100	100 × 10	100 × 25	100 × 50	100 × 100
ARAS	0.0003 ± 0.0000	0.0007 ± 0.0001	0.0011 ± 0.0000	0.0051 ± 0.0007	0.0004 ± 0.0000	0.0011 ± 0.0000	0.0048 ± 0.0006	0.0036 ± 0.0001
CoCoSo	0.0002 ± 0.0000	0.0003 ± 0.0000	0.0006 ± 0.0000	0.0025 ± 0.0005	0.0002 ± 0.0000	0.0004 ± 0.0000	0.0013 ± 0.0002	0.0013 ± 0.0000
CODAS	0.0024 ± 0.0001	0.0027 ± 0.0000	0.0031 ± 0.0000	0.0093 ± 0.0006	0.0087 ± 0.0001	0.0091 ± 0.0001	0.0233 ± 0.0009	0.0127 ± 0.0001
COPRAS	0.0001 ± 0.0000	0.0001 ± 0.0000	0.0001 ± 0.0000	0.0002 ± 0.0002	0.0001 ± 0.0000	0.0001 ± 0.0000	0.0002 ± 0.0002	0.0001 ± 0.0000
CRADIS	0.0001 ± 0.0000	0.0002 ± 0.0000	0.0004 ± 0.0000	0.0014 ± 0.0004	0.0001 ± 0.0000	0.0002 ± 0.0000	0.0013 ± 0.0004	0.0007 ± 0.0000
EDAS	0.0009 ± 0.0000	0.0021 ± 0.0000	0.0042 ± 0.0001	0.0141 ± 0.0008	0.0017 ± 0.0000	0.0038 ± 0.0001	0.0179 ± 0.0009	0.0166 ± 0.0002
GRA	0.0002 ± 0.0000	0.0004 ± 0.0000	0.0009 ± 0.0000	0.0034 ± 0.0006	0.0002 ± 0.0000	0.0004 ± 0.0000	0.0021 ± 0.0005	0.0017 ± 0.0000
MABAC	0.0103 ± 0.0001	0.0265 ± 0.0002	0.0540 ± 0.0003	0.2053 ± 0.0031	0.0320 ± 0.0002	0.0790 ± 0.0004	0.1955 ± 0.0068	0.3157 ± 0.0019
MACBETH	0.0001 ± 0.0000	0.0002 ± 0.0000	0.0004 ± 0.0000	0.0016 ± 0.0004	0.0001 ± 0.0000	0.0002 ± 0.0000	0.0004 ± 0.0000	0.0008 ± 0.0000
MAIRCA	0.0003 ± 0.0000	0.0006 ± 0.0001	0.0011 ± 0.0000	0.0046 ± 0.0005	0.0002 ± 0.0000	0.0006 ± 0.0000	0.0012 ± 0.0000	0.0022 ± 0.0000
MARCOS	0.0001 ± 0.0000	0.0003 ± 0.0000	0.0005 ± 0.0000	0.0023 ± 0.0005	0.0002 ± 0.0000	0.0003 ± 0.0000	0.0006 ± 0.0001	0.0010 ± 0.0000
MAUT	0.0008 ± 0.0000	0.0021 ± 0.0000	0.0040 ± 0.0001	0.0179 ± 0.0009	0.0014 ± 0.0001	0.0031 ± 0.0000	0.0061 ± 0.0001	0.0119 ± 0.0001
MOORA	0.0001 ± 0.0000	0.0002 ± 0.0000	0.0002 ± 0.0000	0.0008 ± 0.0002	0.0001 ± 0.0000	0.0002 ± 0.0000	0.0003 ± 0.0000	0.0004 ± 0.0000
MOOSRA	0.0001 ± 0.0000	0.0002 ± 0.0000	0.0002 ± 0.0000	0.0015 ± 0.0004	0.0001 ± 0.0000	0.0002 ± 0.0000	0.0003 ± 0.0000	0.0004 ± 0.0000
MULTIMOORA	0.0003 ± 0.0000	0.0011 ± 0.0000	0.0019 ± 0.0000	0.0066 ± 0.0007	0.0004 ± 0.0000	0.0016 ± 0.0000	0.0034 ± 0.0001	0.0048 ± 0.0001
OCRA	0.0002 ± 0.0000	0.0007 ± 0.0001	0.0011 ± 0.0000	0.0049 ± 0.0005	0.0003 ± 0.0000	0.0008 ± 0.0000	0.0016 ± 0.0001	0.0030 ± 0.0000
ORESTE	0.0015 ± 0.0000	0.0041 ± 0.0001	0.0073 ± 0.0001	0.0311 ± 0.0012	0.0024 ± 0.0001	0.0060 ± 0.0001	0.0117 ± 0.0001	0.0226 ± 0.0001
PIV	0.0001 ± 0.0000	0.0004 ± 0.0000	0.0006 ± 0.0000	0.0023 ± 0.0001	0.0001 ± 0.0000	0.0003 ± 0.0000	0.0007 ± 0.0000	0.0012 ± 0.0000
PROMETHEE II	0.0268 ± 0.0003	0.0686 ± 0.0006	0.1572 ± 0.0042	0.3334 ± 0.0089	0.1826 ± 0.0051	0.2800 ± 0.0069	0.5142 ± 0.0036	0.9437 ± 0.0081
PROMETHEE IV	0.0954 ± 0.0012	0.2359 ± 0.0025	0.8862 ± 0.0132	1.8584 ± 0.0367	0.7867 ± 0.0102	1.7114 ± 0.0446	2.4003 ± 0.0721	4.1288 ± 0.0785
EC PROMETHEE	0.0859 ± 0.0050	0.4551 ± 0.0057	0.8928 ± 0.0083	1.4930 ± 0.0488	0.6465 ± 0.0177	1.5108 ± 0.0419	2.9664 ± 0.0822	6.1946 ± 0.1198
PSI	0.0001 ± 0.0000	0.0002 ± 0.0000	0.0004 ± 0.0000	0.0006 ± 0.0000	0.0002 ± 0.0002	0.0003 ± 0.0000	0.0003 ± 0.0000	0.0009 ± 0.0000
ROV	0.0002 ± 0.0000	0.0006 ± 0.0000	0.0016 ± 0.0001	0.0023 ± 0.0001	0.0006 ± 0.0003	0.0007 ± 0.0000	0.0011 ± 0.0000	0.0028 ± 0.0001
SAW	0.0001 ± 0.0000	0.0002 ± 0.0000	0.0007 ± 0.0001	0.0007 ± 0.0000	0.0000 ± 0.0000	0.0003 ± 0.0000	0.0004 ± 0.0000	0.0011 ± 0.0001
SPOTIS	0.0000 ± 0.0000	0.0000 ± 0.0000	0.0001 ± 0.0000	0.0001 ± 0.0000	0.0000 ± 0.0000	0.0000 ± 0.0000	0.0000 ± 0.0000	0.0002 ± 0.0000
TODIM	0.1044 ± 0.0006	0.2643 ± 0.0010	0.8447 ± 0.0255	1.5841 ± 0.0352	0.6662 ± 0.0201	2.0244 ± 0.0315	2.0392 ± 0.0193	4.9159 ± 0.0870
TOPSIS	0.0001 ± 0.0000	0.0002 ± 0.0000	0.0006 ± 0.0003	0.0007 ± 0.0000	0.0001 ± 0.0000	0.0005 ± 0.0002	0.0004 ± 0.0000	0.0008 ± 0.0000
VIKOR	0.0002 ± 0.0000	0.0006 ± 0.0001	0.0014 ± 0.0004	0.0010 ± 0.0000	0.0003 ± 0.0000	0.0008 ± 0.0003	0.0005 ± 0.0001	0.0009 ± 0.0000
WSM	0.0002 ± 0.0000	0.0004 ± 0.0000	0.0019 ± 0.0004	0.0017 ± 0.0001	0.0002 ± 0.0000	0.0010 ± 0.0002	0.0008 ± 0.0000	0.0017 ± 0.0001
WPM	0.0002 ± 0.0000	0.0004 ± 0.0000	0.0019 ± 0.0004	0.0017 ± 0.0001	0.0002 ± 0.0000	0.0010 ± 0.0002	0.0008 ± 0.0000	0.0017 ± 0.0001
WASPAS	0.0002 ± 0.0000	0.0004 ± 0.0000	0.0019 ± 0.0004	0.0017 ± 0.0001	0.0002 ± 0.0000	0.0010 ± 0.0002	0.0008 ± 0.0000	0.0017 ± 0.0001

Source(s): Authors’ own creation

the latest advancements in the field. Furthermore, its integration with ChatGPT offers an additional layer of AI-powered analysis, although researchers should exercise caution and discernment when interpreting outputs. Notably, PyDecision has modules that facilitate comparisons between MCDA methods, whether those generating ranks or weights. In addition, its voting methods are adapted for MCDA contexts, enabling the selection of top alternatives from rank-comparison matrices. The fusion of MCDA with AI through PyDecision modernizes both time-tested and novel approaches, exemplifying the future of decision-making tools.

In future work, we intend to conduct controlled experiments and comparative studies to systematically evaluate ChatGPT's integration within decision-making processes. These studies will aim to quantify the impact of AI-driven interpretations on key factors such as decision accuracy, decision-making speed and usability. These empirical evaluations will provide deeper insights into the practical benefits and limitations of incorporating LLM into decision analysis frameworks like pyDecision, contributing to the development of more robust, adaptive and data-driven decision support systems.

Declarations

No funding was received to assist with the preparation of this manuscript.

All authors certify that they have no affiliations with or involvement in any organization or entity with any financial interest or nonfinancial interest in the subject matter or materials discussed in this manuscript.

The data sets and computational details for the research presented are fully accessible to ensure transparency and reproducibility. The data for the first case study can be found at https://bit.ly/47aAVrZ, and the data for the second case study is available at https://bit.ly/3QAaJle

References

Amiri, M., Hashemi-Tabatabaei, M., Ghahremanloo, M., Keshavarz-Ghorabaee, M., Zavadskas, E.K. and Kaklauskas, A. (2021), "Evaluating life cycle of buildings using an integrated approach based on Quantitative-Qualitative and simplified Best-Worst methods (QQM-SBWM)", *Sustainability*, Vol. 13 No. 8, p. 4487, doi: 10.3390/su13084487.

Basilio, M.P., Pereira, V. and Yigit, F. (2023), "New hybrid EC-Promethee method with multiple iterations of random weight ranges: APPLIED to the choice of policing strategies", *Mathematics*, Vol. 11 No. 21, p. 11214432, doi: 10.3390/math11214432.

Bigaret, S., Hodgett, R.E., Meyer, P., Mironova, T. and Olteanu, A.L. (2017), "Supporting the multicriteria decision aiding process: r and the MCDA package", *EURO Journal on Decision Processes*, Vol. 5 Nos 1-4, pp. 169-194, doi: 10.1007/s40070-017-0064-1.

Bisdorf, R., Dias, L., Mousseau, V., Pirlot, M. and Meyer, P. (2015), "Evaluation and decision models with multiple criteria. Case studies", *International Handbooks on Information Systems*, Springer, Berlin, doi: 10.1007/978-3-662-46816-6.

Borda, J.C. (1781), "Mémoire sur les élections au scrutin", *Histoire De L'Académie Royale Des Sciences*, Vol. 102, pp. 657-665, available at: http://gerardgreco.free.fr/IMG/pdf/MA_c_moire-Borda-1781.pdf

Bottero, M., Ferretti, V., Figueira, J.R., Greco, S. and Roy, B. (2015), "Dealing with a multiple criteria environmental problem with interaction effects between criteria through an extension of the ELECTRE III method", *European Journal of Operational Research*, Vol. 245 No. 3, pp. 837-850, doi: 10.1016/j.ejor.2015.04.005.

Bouyssou, D. and Marchant, T. (2007), "An axiomatic approach to noncompensatory sorting methods in MCDM, II: more than two categories", *European Journal of Operational Research*, Vol. 178 No. 1, pp. 246-276, doi: 10.1016/j.ejor.2006.01.033.

Brans, J.P. and Vincke, P. (1985), "A preference ranking organisation method", *Management Science*, Vol. 31 No. 6, pp. 647-656, doi: 10.1287/mnsc.31.6.647.

Brauers, W.K. and Zavadskas, E.K. (2006), "The MOORA method and its application to privatization in a transition economy", *Control and Cybernetics*, Vol. 35 No. 2, pp. 445-469.

Brauers, W.K. and Zavadskas, E.K. (2010), "Project management by multimoora as an instrument for transition economies", *Technological and Economic Development of Economy*, Vol. 16 No. 1, pp. 5-24, doi: 10.3846/tede.2010.01.

Churchman, C.W., Ackoff, R.L. and Smith, N.M. (1954), "An approximate measure of value", *Journal of the Operations Research Society of America*, Vol. 2 No. 2, pp. 172-187, doi: 10.1287/opre.2.2.172.

Cinelli, M., Coles, S.R. and Kirwan, K. (2014), "Analysis of the potentials of multicriteria decision analysis methods to conduct sustainability assessment", *Ecological Indicators*, Vol. 46, pp. 138-148, doi: 10.1016/j.ecolind.2014.06.011.

Cinelli, M., Spada, M., Kim, W., Zhang, Y. and Burgherr, P. (2021), "MCDA index tool: an interactive software to develop indices and rankings", *Environment Systems and Decisions*, Vol. 41 No. 1, pp. 82-109, doi: 10.1007/s10669-020-09784-x.

Copeland, A.H. (1951), "A reasonable social welfare function", Mimeographed notes from a Seminar on Applications of Mathematics to the Social Sciences.

Costa, C.A.B. and Vansnick, J.C. (1994), "MACBETH – an interactive path towards the construction of cardinal value functions", *International Transactions in Operational Research*, Vol. 1 No. 4, pp. 489-500, doi: 10.1016/0969-6016(94)90010-8.

Creative Decisions Foundation (2020), "SuperDecisions [software]", available at: www.superdecisions.com

Dehghan-Manshadi, B., Mahmudi, H., Abedian, A. and Mahmudi, R. (2007), "A novel method for materials selection in mechanical design: combination of non-linear normalization and a modified digital logic method", *Materials and Design*, Vol. 28 No. 1, pp. 8-15, doi: 10.1016/j.matdes.2005.06.023.

Deng, J.L. (1989), "Introduction to grey system theory", *The Journal of Grey System*, Vol. 1 No. 1, pp. 1-24, available at: https://uranos.ch/research/references/Julong_1989/10.1.1.678.3477.pdf

Dezert, J., Tchamova, A., Han, D. and Tacnet, J.M. (2020), "The SPOTIS rank reversal free method for multicriteria Decision-Making support", *2020 IEEE 23rd International Conference on Information Fusion (FUSION)*, doi: 10.23919/FUSION45008.2020.9190347.

Diakoulaki, D., Mavrotas, G. and Papayannakis, L. (1995), "Determining objective weights in multiple criteria problems: the CRITIC method", *Computers and Operations Research*, Vol. 22 No. 7, pp. 763-770, doi: 10.1016/0305-0548(94)00059-H.

Edwards, W. (1971), "Social utilities. Engineering economist", *Engineering Economic Systems*, Vol. 16 No. 3, pp. 119-129.

Enterprise Software (2023), "D-Sight [software]", available at: www.d-sight.com

Expertchoice (2020), "ExpertChoice [software]", ExpertChoice Solutions, available at: www.expertchoice.com

Fishburn, P.C. (1967), "Additive utilities with incomplete product set: applications to priorities and assignments", *Journal of the Operations Research Society of America*, Vol. 15 No. 3, doi: 10.1287/opre.15.3.537.

Floriano, C.M., Pereira, V. and Rodrigues, B.E.S. (2022), "3MO-AHP: an inconsistency reduction approach through Mono-, multi- or many-objective quality measures", *Data Technologies and Applications*, Vol. 56 No. 5, pp. 645-670, doi: 10.1108/DTA-11-2021-0315.

Fontela, E. and Gabus, A. (1976), *The DEMATEL Observer*, Battelle Geneva Research Centre, Geneva.

Fu, Y.K., Wu, C.J. and Liao, C.N. (2021), "Selection of In-Flight Duty-Free product suppliers using a combination fuzzy AHP, fuzzy ARAS, and MSGP methods", *Mathematical Problems in Engineering*, Vol. 2021, p. 8545379, doi: 10.1155/2021/8545379.

Ghorabaee, M.K., Zavadskas, E.K., Olfat, L. and Turskis, Z. (2015), "Multi-criteria inventory classification using a new method of evaluation based on distance from average solution (EDAS)", *Informatica*, Vol. 26 No. 3, pp. 435-451, doi: 10.15388/Informatica.2015.57.

Ghorabaee, M.K., Zavadskas, E.K., Turskis, Z. and Antucheviciene, J. (2016), "A new combinative Distance-Based assessment(codas) method for multicriteria Decision-Making", *Economic Computation and Economic Cybernetics Studies and Research*, Vol. 50 No. 3, pp. 25-44, available at: https://ideas.repec.org/a/cys/ecocyb/v50y2016i3p25html

Gomes, L.F.A.M. and Rangel, L.A.D. (2009), "An application of the TODIM method to the multicriteria rental evaluation of residential properties", *European Journal of Operational Research*, Vol. 193 No. 1, pp. 204-211, doi: 10.1016/j.ejor.2007.10.046.

Greco, S., Ehrgott, M. and Figueira, J. (2016), *Multiple Criteria Decision Analysis: State of the Art Surveys*, Springer, New York, NY, doi: 10.1007/b100605.

Hansen, P. and Ombler, F. (2008), "A new method for scoring additive multi-attribute value models using pairwise rankings of alternatives", *Journal of Multi-Criteria Decision Analysis*, Vol. 15 Nos 3-4, pp. 87-107, doi: 10.1002/mcda.428.

Heymann, M.C., Pereira, V. and Caiado, R.G.G. (2024), "PyMissingAHP: an evolutionary algorithm for filling missing values in incomplete pairwise comparisons matrices with real or fuzzy numbers via Mono and multiobjective approaches", *Arabian Journal for Science and Engineering*, Vol. 49 No. 5, pp. 7375-7394, doi: 10.1007/s13369-023-08227-4.

Hinloopen, E. and Nijkamp, P. (1990), "Qualitative multiple criteria choice analysis", *Quality and Quantity*, Vol. 24 No. 1, pp. 37-56, doi: 10.1007/BF00221383.

Hodgett, R.E., Martin, E.B., Montague, G. and Talford, M. (2014), "Handling uncertain decisions in whole process design", *Production Planning and Control*, Vol. 25 No. 12, pp. 1028-1038, doi: 10.1080/09537287.2013.798706.

Hodgett, R.E. and Siraj, S. (2019), "SURE: a method for decision-making under uncertainty", *Expert Systems with Applications*, Vol. 115, pp. 684-694, doi: 10.1016/j.eswa.2018.08.048.

Hwang, C.L. and Yoon, K. (1981), *Multiple Attribute Decision Making: Methods and Applications*, Springer-Verlag, Cham, doi: 10.1007/978-3-642-48318-9.

INFOHARVEST (2023), "Criterium DecisionPlus [software]", available at: www.infoharvest.com/ihroot/infoharv/cdpfreedownloads.asp

Jacquet-Lagreze, E. and Siskos, J. (1982), "Assessing a set of additive utility functions for multicriteria decision-making, the UTA method", *European Journal of Operational Research*, Vol. 10 No. 2, pp. 151-164, doi: 10.1016/0377-2217(82)90155-2.

Jagadish, J. and Ray, A. (2014), "Green cutting fluid selection using MOOSRA method", *International Journal of Research in Engineering and Technology*, Vol. 3 No. 15, doi: 10.15623/ijret.2014.0315105.

Kalita, K., Pal, S. and Haldar, S. (2022), "A hybrid TOPSIS-PR-GWO approach for multi-objective process parameter optimization", *Process Integration and Optimization for Sustainability*, Vol. 6 No. 4, pp. 1011-1026, doi: 10.1007/s41660-022-00256-0.

Keeney, R.L. and Raiffa, H. (1976), *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*, Wiley, London, ISBN-10: 0521438837.

Keshavarz-Ghorabaee, M., Amiri, M., Zavadskas, E.K., Turskis, Z. and Antucheviciene, J. (2021), "Determination of objective weights using a new method based on the removal effects of criteria (MEREC)", *Symmetry*, Vol. 13 No. 4, p. sym13040525, doi: 10.3390/sym13040525.

Keshavarz-Ghorabaee, M., K.G., Amiri, M., Zavadskas, E.K., Turskis, Z. and Antucheviciene, J. (2018), "Simultaneous evaluation of criteria and alternatives (SECA) for multicriteria Decision-Making", *Informatica*, Vol. 29 No. 2, pp. 265-280, doi: 10.15388/Informatica.2018.167.

Khannoussi, A., Olteanu, A.L., Labreuche, C. and Meyer, P. (2022), "Simple ranking method using reference profiles: incremental elicitation of the preference parameters", *4OR*, Vol. 20 No. 3, pp. 499-530, doi: 10.1007/s10288-021-00487-w.

Kim, J.K., Chua, M., Rickard, M. and Lorenzo, A. (2023), "ChatGPT and large language model (LLM) chatbots: the current state of acceptability and a proposal for guidelines on utilization in academic medicine", *Journal of Pediatric Urology*, Vol. 19 No. 5, p. 37328321, doi: 10.1016/j.jpurol.2023.05.018.

Kizielewicz, B., Shekhovtsov, A. and Sałabun, W. (2023), "PYMCDN – The universal library for solving multicriteria decision-making problems", *SoftwareX*, Vol. 22, p. 101368, doi: 10.1016/j.softx.2023.101368.

Madić, M., Radovanović, M. and Manić, M. (2016), "Application of the ROV method for the selection of cutting fluids", *Decision Science Letters*, Vol. 5, pp. 245-254, doi: 10.5267/j.dsl.2015.12.001.

Maniya, K. and Bhatt, M.G. (2010), "A selection of material using a novel type decision-making method: preference selection index method", *Materials and Design*, Vol. 31 No. 4, pp. 1785-1789, doi: 10.1016/j.matdes.2009.11.020.

Manshadi, B.D., Mahmudi, H., Abedian, A. and Mahmudi, R. (2007), "A novel method for materials selection in mechanical design: combination of non-linear normalization and a modified digital logic method", *Materials and Design*, Vol. 28 No. 1, pp. 8-15, doi: 10.1016/j.matdes.2005.06.023.

Michnik, J. (2013), "Weighted influence non-linear gauge system (WINGS) – an analysis method for the systems of interrelated components", *European Journal of Operational Research*, Vol. 228 No. 3, pp. 536-544, doi: 10.1016/j.ejor.2013.02.007.

Miller, D.W. and Starr, M.K. (1960), *Executive Decisions and Operations Research*, Prentice-Hall, Inc, Englewood Cliffs, NJ, U.S.A.

Miloš, M., Dušan, P. and Miroslav, R. (2015), "Selection of non-conventional machining processes using the OCRA method", *Serbian Journal of Management*, Vol. 10 No. 1, pp. 61-73, doi: 10.5937/sjm10-6802.

MINDS (2023), "1000minds decision-making and conjoint analysis software [software]", available at: www.1000minds.com

Montenegro DE Barros, G.M., Pereira, V. and Roboredo, M.C. (2021), "ELECTRE tree: a machine learning approach to infer ELECTRE Tri-B parameters", *Data Technologies and Applications*, Vol. 55 No. 4, pp. 586-608, doi: 10.1108/DTA-10-2020-0256.

Mufazzal, S. and Muzakkir, S.M. (2018), "A new Multi-Criterion decision making (MCDM) method based on proximity indexed value for minimizing rank reversals", *Computers and Industrial Engineering*, Vol. 119, pp. 427-438, doi: 10.1016/j.cie.2018.03.045.

Narang, M., Joshi, M.C. and Pal, A.K. (2021), "A hybrid fuzzy COPRAS-base-criterion method for multicriteria decision making", *Soft Computing*, Vol. 25 No. 13, pp. 8391-8399, doi: 10.1007/s00500-021-05762-w.

OPENAI (2023), "GPT-4 technical report", arXiv, doi: 10.48550/arXiv.2303.08774.

Opricović, S. (1990), "Programski paket VIKOR za visekriterijumsko kompromisno rangiranje", *Proceedings of the 17th International Symposium on Operational Research SYM-OP-IS*; *Dubrovnik-Kupari, Croatia*. 9 – 12 October.

Pamučar, D. and S., Ćirović, G. (2015), "The selection of transport and handling resources in logistics centers using Multi-Attributive border approximation area comparison (MABAC)", *Expert Systems with Applications*, Vol. 42 No. 6, pp. 3016-3028, doi: 10.1016/j.eswa.2014.11.057.

Pamučar, D.S., Tarle, S.P. and Parezanovic, T. (2018), "New hybrid multicriteria decision-making DEMATEL-MAIRCA model: sustainable selection of a location for the development of multimodal logistics Centre", *Economic Research-Ekonomska Istraživanja*, Vol. 31 No. 1, pp. 1641-1665, doi: 10.1080/1331677X.2018.1506706.

Pendokhare, D., Kalita, K., Chakraborty, S. and Čep, R. (2024), "A comprehensive review of parametric optimization of electrical discharge machining processes using multi-criteria decision-making techniques", *Frontiers in Mechanical Engineering*, Vol. 10, p. 1404116, doi: 10.3389/fmech.2024.1404116.

Pereira, V. (2022), “Voracious-AHP”, Github, available at: https://github.com/Valdecy/Voracious-AHP

Pereira, V. and Nepomuceno, L., D.O. (2021), “J-ELECTRE. Sourceforge”, available at: https://sourceforge.net/projects/jlectre/files

Pereira, F.C., Costa, H.G. and Pereira, V. (2017), “Patent filings versus articles published: a review of the literature in the context of multicriteria decision aid”, *World Patent Information*, Vol. 50, pp. 17-26, doi: 10.1016/j.wpi.2017.07.003.

Puška, A., Stević, Ž. and Pamučar, D.S. (2022), “Evaluation and selection of healthcare waste incinerators using extended sustainability criteria and multicriteria analysis methods”, *Environment, Development and Sustainability*, Vol. 24 No. 9, pp. 11195-11225, doi: 10.1007/s10668-021-01902-2.

Rao, R.V. (2006), “A material selection model using graph theory and matrix approach”, *Materials Science and Engineering: A*, Vol. 431 Nos 1-2, pp. 248-255, doi: 10.1016/j.msea.2006.06.006.

Rezaei, J. (2015), “Best-worst multicriteria decision-making method”, *Omega*, Vol. 53, pp. 49-57, doi: 10.1016/j.omega.2014.11.009.

Rodrigues, B.S., Floriano, C.M., Pereira, V. and Roboredo, M.C. (2021), “An algorithm to elicitate Electre II, III and IV parameters”, *Data Technologies and Applications*, Vol. 55 No. 1, pp. 82-96, doi: 10.1108/DTA-07-2020-0161.

Roubens, M. (1982), “Preference relations on actions and criteria in multicriteria decision making”, *European Journal of Operational Research*, Vol. 10 No. 1, pp. 51-55, doi: 10.1016/0377-2217(82)90131-X.

Roy, B. (1968), “Classement et choix en présence de points de vue multiples (la méthode ELECTRE)”, *Revue Française d’;Informatique et de Recherche Opérationnelle*, Vol. 2 No. 8, pp. 57-75.

Saaty, T.L. (1990), “How to make a decision: the analytic hierarchy process”, *European Journal of Operational Research*, Vol. 48 No. 1, pp. 9-26, doi: 10.1016/0377-2217(90)90057-I.

Sałabun, W. (2015), “The characteristic objects method: a new distance-based approach to multicriteria decision-making problems”, *Journal of Multi-Criteria Decision Analysis*, Vol. 22 Nos 1-2, pp. 37-50, doi: 10.1002/mcda.1525.

Satman, M.H., Yildirim, B.F. and Kuruca, E. (2021), “JMcDM: a Julia package for Multiple-Criteria Decision-Making tools”, *Journal of Open Source Software*, Vol. 6 No. 65, p. 65, doi: 10.21105/joss.03430.

Shannon, C.E. (1948), “A mathematical theory of communication”, *Bell System Technical Journal*, Vol. 27 No. 3, pp. 379-423, doi: 10.1002/j.1538-7305.1948.tb01338.x.

Sommerville, I. (2011), *Software Engineering*, 9th ed., Pearson, London, ISBN-13: 9780137035151. ISBN-10: 0137035152.

Stević, Z., Pamučar D, S., Puška, A. and Prasenjit Chatterjee, P. (2020), “Sustainable supplier selection in healthcare industries using a new MCDM method: measurement of alternatives and ranking according to compromise solution (MARCOS)”, *Computers and Industrial Engineering*, Vol. 140, p. 106231, doi: 10.1016/j.cie.2019.106231.

Taherdoost, H. and Madanchian, M. (2023), “Multicriteria decision making (MCDM) methods and concepts”, *Encyclopedia*, Vol. 3 No. 1, pp. 77-87, doi: 10.3390/encyclopedia3010006.

Trivedi, P., Shah, J., Čep, R., Abualigah, L. and Kalita, K. (2024), “A hybrid Best-Worst method (BWM) -Technique for order of preference by similarity to ideal solution (TOPSIS) approach for prioritizing road safety improvements”, *IEEE Access*, Vol. 12, pp. 30054-30065, doi: 10.1109/ACCESS.2024.3368395.

Wątróbski, J., Bączkiewicz, A. and Sałabun, W. (2022), “ Pyrepo-mcda - Reference objects based MCDA software package”, *SoftwareX*, Vol. 19, p. 101107, doi: 10.1016/j.softx.2022.101107.

Więckowski, J., Kizielewicz, B. and Sałabun, W. (2023a), “pyFDM: a python library for uncertainty decision analysis methods”, *SoftwareX*, Vol. 20, p. 101271, doi: 10.1016/j.softx.2022.101271.

Więckowski, J., Kizielewicz, B. and Sałabun, W. (2023b), "Handling decision-making in intuitionistic fuzzy environment: PYIFDM package", *SoftwareX*, Vol. 22, pp. 101344, doi: 10.1016/j.softx.2023.101344.

Yazdani, M., Zarate, P., Zavadskas, E.K. and Turskis, Z. (2019), "A combined compromise solution (CoCoSo) method for multicriteria decision-making problems", *Management Decision*, Vol. 57 No. 9, pp. 2501-2519, doi: 10.1108/MD-05-2017-0458.

Zavadskas, E.K. and Turskis, Z. (2010), "A new additive ratio assessment (ARAS) method in multicriteria decision-making", *Technological and Economic Development of Economy*, Vol. 16 No. 2, pp. 159-172, doi: 10.3846/tede.2010.10.

Zavadskas, E.K. and Podvezko, V. (2016), "Integrated determination of objective criteria weights in MCDM", *International Journal of Information Technology and Decision Making*, Vol. 15 No. 2, pp. 267-283, doi: 10.1142/S0219622016500036.

Zavadskas, E.K., Kaklauskas, A. and Sarka, V. (1994), "The new method of multicriteria complex proportional assessment of projects", *Technological and Economic Development of Economy*, Vol. 1 No. 3, pp. 131-139, available at: www.researchgate.net/publication/285902199_The_new_method_of_multicriteria_complex_proportional_assessment_of_projects

Zavadskas, E.K., Vilutienė, T., Turskis, Z. and Šaparauskas, J. (2014), "Multicriteria analysis of projects' performance in construction", *Archives of Civil and Mechanical Engineering*, Vol. 14 No. 1, pp. 114-121, doi: 10.1016/j.acme.2013.07.006.

Ziemba, P. (2020), "Multicriteria stochastic selection of electric vehicles for the sustainable development of local government and state administration units in Poland", *Energies*, Vol. 13 No. 23, doi: 10.3390/en13236299.

Zindani, D., Maity, S.R. and Bhowmik, S. (2019), "Fuzzy-EDAS (evaluation based on distance from average solution) for material selection problems", in Narayanan, R., Joshi, S., Dixit, U. (Eds), *Advances in Computational Methods in Manufacturing. Lecture Notes on Multidisciplinary Industrial Engineering*, Springer, Singapore, doi: 10.1007/978-981-32-9072-3_63.

Further reading

Turskis, Z. and Zavadskas, E.K. (2010), "A new fuzzy additive ratio assessment method (aras-F). Case study: the analysis of fuzzy multiple criteria in order to select the logistic centers location", *TRANSPORT*, Vol. 25 No. 4, pp. 423-432, doi: 10.3846/transport.2010.52.

Van Laarhoven, P.J.M. and Pedrycz, W. (1983), "A fuzzy extension of Saaty's priority theory", *Fuzzy Sets and Systems*, Vol. 11 Nos 1-3, pp. 229-241, doi: 10.1016/S0165-0114(83)80082-7.

Corresponding author
Valdecy Pereira can be contacted at: valdecy.pereira@gmail.com