JMcDM: A Julia package for multiple-criteria decision making tools

Mehmet Hakan Satman[a], Bahadır Fatih Yıldırım[b], Ersagun Kuruca[c]

[a]*Istanbul University, Department of Econometrics, Beyazit, Istanbul, Turkey*
[b]*Istanbul University, Department of Transportation and Logistics, Avcilar, Istanbul, Turkey*
[c]*Istanbul Technical University, Department of Computer Engineering, Sariyer, Istanbul, Turkey*

Abstract

Ca. 100 words

Keywords: keyword 1, keyword 2, keyword 3

Required Metadata

Current code version

Ancillary data table required for subversion of the codebase. Kindly replace examples in right column with the correct information about your current code, and leave the left column as it is.

Nr.	Code metadata description	Please fill in this column
C1	Current code version	v0.1.5
C2	Permanent link to code/repository used for this code version	https://github.com/jbytecode/JMcDM
C3	Code Ocean compute capsule	
C4	Legal Code License	MIT
C5	Code versioning system used	git
C6	Software code languages, tools, and services used	Julia
C7	Compilation requirements, operating environments & dependencies	Julia 1.4
C8	If available Link to developer documentation/manual	https://jbytecode.github.io/JMcDM/docs/build
C9	Support email for questions	mhsatman@istanbul.edu.tr

Table 1: Code metadata (mandatory)

1. Motivation and significance

Introduce the scientific background and the motivation for developing the software. Explain why the software is important, and describe the exact (scientific) problem(s) it Indicate in what way the software has contributed (or how it will contribute in the future) to Provide a description of the experimental setting (how does the user use the software?). Introduce related work in literature (cite or list algorithms used, other software etc.).

The one-dimensional array a is in ascending order if and only if $a_i \leq a_{i+1}$

where $i = 1, 2, \ldots, n-1$, and n is the length of array. In other terms, the process of ordering numbers requires the logical $\leq$ operator to be perfectly defined. Since the operator $\leq$ is not defined for any set of points in higher dimensions, $\mathbb{R}^p$ for $p \geq 2$, there is not a unique ordering of points.

In multi-dimensional case, the binary domination operator $\succ$ applied on points a and b, $a \succ b$, is true iif each item in a is not worse than the correspoing item in b and at least one item is better than the corresponding item in b (1). On the other hand, the more relaxed operator $\succeq$ returns true if each item in a is as good as the corresponding item in b (2). Several outranking methdods in MCDA (Multiple-Criteria Decision Analysis) define a unique ranking mechanism to select the best alternative among others.

Suppose a decision process has n alternatives and m criteria which are either to be maximized or minimized. Each single criterion has a weight $0 \leq w_i \leq 1$ where $\sum_i^m w_i = 1$. f_i is either maximum or minimum. $g_j(.)$ is evolution function and it is taken as $g_j(x) = x$ in many methods. A multiple criteria decision problem can be represented using the decision table

Criteria	C_1	C_2	$\ldots$	C_m
Weights	w_1	w_2	$\ldots$	w_m
Functions	f_1	f_2	$\ldots$	f_m
A_1	$g_1(A_1)$	$g_2(A_1)$	$\ldots$	$g_m(S_A)$
A_2	$g_1(A_2)$	$g_2(A_2)$	$\ldots$	$g_m(A_2)$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
A_n	$g_1(A_n)$	$g_2(A_n)$	$\ldots$	$g_m(A_n)$

Table 2: Sample decision matrix

without loss of generality. When A_1, A_2, ..., A_n are alternatives and C_1, C_2, ..., C_m are different situations of a single criterion then the decision problem is said to be single criterion decision problem. If A_i and C_j are strategies of two game players then $g_j(A_i)$ is the gain of the row player when she selects the strategy i and the column player selects the strategy C_j.

We must put here some text here like: MCDA is used in location selection of facilities (Ref), selection of suppliers (Ref).... .

Multiple-criteria decision-making (MCDM) tools provide several algorithms for ordering or selecting alternatives and/or determining the weigths when there is uncertainity. Although some algorithms are suitable for hand calculations, a computer software is often required. *PyTOPS* is a Python tool for TOPSIS (3). *Super Decisions* is a software package which is mainly focused on AHP (Analytic Hierarchy Process) and ANP (Analytic Network Process) (4). *Visual Promethee* implements Promethee method on Windows platforms (5). *M-BACBETH* is an other commericial software product that implements MACBETH with a easy to use GUI. (6). [List more software here if exist.]

JMcDM is designed to provide a developer-friendly library for solving multiple-criteria decision problems in Julia (7). Since Julia is a dynamic language, it is also useful for researchers that familiar with REPL environments. The package includes multi-criteria decision methods as well as a game solver for zero-sum games and methods for single criterion methods.

2. Software description

2.1. Software Architecture

Give a short overview of the overall software architecture; provide a pictorial component overview or similar (if possible). If necessary provide implementation details.

JMcDM provides a framework for performing multi-criteria decision analysis as well as it includes utility functions for development of new methods. Each single MCDM method returns an object in subtype of `MCDMResult` which is defined as

```
abstract type MCDMResult end
```

and it is used to derive new return types. For instance, the `topsis()` function always returns a `TopsisResult` object which is defined as

```
struct TopsisResult <: MCDMResult
    decisionMatrix::DataFrame
    weights::Array{Float64,1}
    normalizedDecisionMatrix::DataFrame
    normalizedWeightedDecisionMatrix::DataFrame
    bestIndex::Int64
    scores::Array{Float64,1}
end
```

and holds many outputs in a single `struct`. Function definitions are also similar but they may differ depending on the requirements of algorithms. For instance the function `topsis` is defined as

```
function topsis(
    decisionMat::DataFrame,
```

```
    weights::Array{Float64,1},
    fns::Array{Function,1})::TopsisResult
```

where `decisionMat` is the decision matrix, `weights` are weights of criteria, and `fns` is an array of functions (either `minimum` or `maximum`) that determine the optimization directions.

The package is registered in Julia package repository and it is available for downloading and installing using Julia's package manager.

```
julia> using Pkg
julia> Pkg.add("JMcDM")
```

and

```
julia> ]
(@v1.5) pkg> add JMcDM
```

present two distinct ways of install the package.

2.2. Software Functionalities

Present the major functionalities of the software.

The package implements methods for TOPSIS (Technique for Order Preference by Similarity to Ideal Solutions)(8), ELECTRE (Elemination and Choice Translating Reality)(8), PROMETHEE (Preference Ranking Organization METHod for Enrichment of Evaluations)(8), DEMATEL (The Decision Making Trial and Evaluation Laboratory)(8), MOORA (Multi-Objective Optimization By Ratio Analysis)(8), VIKOR (VlseKriterijumska Optimizcija I Kaompromisno Resenje in Serbian)(8), AHP (Analytic Hierarchy Process)(8),

GRA (Grey Relational Analysis)(8), NDS (Non-dominated Sorting)(8), SAW (Simple Additive Weighting)(8), ARAS (Additive Ratio Assessment)(8), WPM (Weighted Product Model)(8), WASPAS (Weighted Aggregated Sum Product ASsessment)(8), EDAS (Evaluation based on Distance from Average Solution)(8), MARCOS (Measurement Alternatives and Ranking according to COmpromise Solution)(8), MABAC (Multi-Attributive Border Approximation area Comparison)(8), MAIRCA (Multi Attributive Ideal-Real Comparative Analysis)(8), COPRAS (COmplex PRoportional ASsessment)(8), COCOSO (Combined Compromise Solution)(8), and CRITIC (CRiteria Importance Through Intercriteria Correlation)(8) for multiple-criteria tools. The package also performs DEA for Data Envelopment Analysis(8) and includes a method for zero-sum game solver.

2.3. Sample code snippets analysis

Suppose a decision problem is given in Table 3.

Criteria	Age	Size	Price	Distance	Population
Weights	0.35	0.15	0.25	0.20	0.05
Functions	min	max	min	min	max
A_1	6	140	150000	950	1500
A_2	4	90	100000	1500	2000
A_3	12	140	75000	550	1100

Table 3: Decision matrix

In this sample problem, a decision maker is subject to select an apartment by considering age of the building, size (in m^2s), price (in \$), distance to city

centre, and nearby population. The data can be entered as a two-dimensional array (matrix) or as a DataFrame object:

```
julia> using JMcDM, DataFrames
julia> df = DataFrame(
:age        => [6.0, 4, 12],
:size       => [140.0, 90, 140],
:price      => [150000.0, 100000, 75000],
:distance   => [950.0, 1500, 550],
:population => [1500.0, 2000, 1100]);

julia> w  = [0.35, 0.15, 0.25, 0.20, 0.05];
julia> fns = [minimum, maximum, minimum, minimum, maximum];
julia> result = topsis(df, w, fns);
julia> result.scores
3-element Array{Float64,1}:
0.5854753145549456
0.6517997936899308
0.41850223305822903

julia> result.bestIndex
2
```

In the output above it is shown that the alternative A_2 has a score of 0.65179 and it is selected as the best. The same analysis can be performed using `saw()` for the method of Simple Additive Weighting

```
julia> result = saw(df, w, fns);
julia> result.bestIndex
2
```

as well as using `wpm` for the method of Weighted Product Method

```
julia> result = wpm(df, w, fns);
julia> result.bestIndex
2
```

For any method, `?methodname` shows the documentation as in the same way in other Julia packages.

3. Illustrative Examples

Provide at least one illustrative example to demonstrate the major functions.

Since *JMcDM* is designed as a software library and for REPL use, it does not implement a significant user interface. However, the `summary()` function provides a useful way to perform a list of methods and returns a text based result to compare results.

```
julia> methods1 = [:topsis, :electre, :vikor,
:moora, :cocoso, :wpm, :waspas]
julia> result1 = summary(df, w, fns, methods1);
```

Figure 1 represents the output of the `summary()` call for methods TOPSIS, ELECTRE, VIKOR, MOORA, COCOSO, WPM, and WASPAS, respectively.

Figure 1: Results of TOPSIS, ELECTRE, VIKOR, MOORA, COCOSO, WPM, and WASPAS

Figure 2: Results of ARAS, SAW, EDAS, MARCOS, MABAC, MAIRCA, and GREY

```
julia> methods2 = [:aras, :saw, :edas, :marcos,
:mabac, :mairca, :grey];
julia> result2 = summary(df, w, fns, methods2);
```

Figure 2 represents the output of the `summary()` call for methods ARAS, SAW, EDAS, MARCOS, MABAC, MAIRCA, and GREY, respectively.

4. Impact

This is the main section of the article and the reviewers weight the description here appropriately Indicate in what way new research

questions can be pursued as a result of the software (if any). Indicate in what way, and to what extent, the pursuit of existing research questions is improved (if so). Indicate in what way the software has changed the daily practice of its users (if so). Indicate how widespread the use of the software is within and outside the intended user group. Indicate in what way the software is used in commercial settings and/or how it led to the creation of spin-off companies (if so).

JMcDM provides a moderate number of MCDA tools and utility functions for developing new methods as well as performing decision analysis using a single function call for each method. A researcher can easly perform sequantial analysis by changing the problem parameters and can compare results of many tools. Existing software packages are mainly focused on providing a small subset of methods. *JMcDM* is an all-in-one solution and has potential for increasing user productivity. Seeing the different results produced by the methods together also helps to discover which parameters the research is more sensitive to and the reasons for them.

5. Conclusions

Set out the conclusion of this original software publication.

6. Conflict of Interest

We wish to confirm that there are no known conflicts of interest associated with this publication and there has been no significant financial support for this work that could have influenced its outcome.

Acknowledgements

The authors would like to thankt he Editor-in-Chief, editors, reviewers for providing extremely insightful comments, and other stuff of the SoftwareX Journal.

References

[1] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Transactions on Evolutionary Computation 6 (2) (2002) 182–197. doi:10.1109/4235.996017.

[2] S. Greco, J. Figueira, M. Ehrgott, Multiple criteria decision analysis, Vol. 37, Springer, 2016.

[3] V. Yadav, S. Karmakar, P. P. Kalbar, A. Dikshit, PyTOPS: A python based tool for TOPSIS, SoftwareX 9 (2019) 217–222. doi:10.1016/j.softx.2019.02.004.

[4] W. Adams, R. Saaty, Super decisions software guide (2003).
URL https://superdecisions.com/sd_resources/v28_man01.pdf

[5] B. Mareschal, Y. D. Smet, Visual PROMETHEE: Developments of the PROMETHEE - GAIA multicriteria decision aid methods, in: 2009 IEEE International Conference on Industrial Engineering and Engineering Management, IEEE, 2009. doi:10.1109/ieem.2009.5373124.

[6] C. A. Bana e Costa, J.-M. de Corte, J.-C. Vansnick, MACBETH (Measuring Attractiveness by a Categorical Based Evaluation Tech-

nique), American Cancer Society, 2011. doi:https://doi.org/10.1002/9780470400531.eorms0970.

[7] J. Bezanson, A. Edelman, S. Karpinski, V. B. Shah, Julia: A fresh approach to numerical computing, SIAM Review 59 (1) (2017) 65–98. doi:10.1137/141000671.
URL https://doi.org/10.1137%2F141000671

[8] C.-L. Hwang, K. Yoon, Methods for multiple attribute decision making, in: Multiple Attribute Decision Making, Springer Berlin Heidelberg, 1981, pp. 58–191. doi:10.1007/978-3-642-48318-9_3.

Current executable software version

Ancillary data table required for sub version of the executable software: (x.1, x.2 etc.) kindly replace examples in right column with the correct information about your executables, and leave the left column as it is.

Nr.	**(Executable) software metadata description**	**Please fill in this column**
S1	Current software version	For example 1.1, 2.4 etc.
S2	Permanent link to executables of this version	For example: *https* : *//github.com/combogenomics/DuctApe/releases/tag/DuctApe* − 0.16.4
S3	Legal Software License	List one of the approved licenses
S4	Computing platforms/Operating Systems	For example Android, BSD, iOS, Linux, OS X, Microsoft Windows, Unix-like , IBM z/OS, distributed/web based etc.
S5	Installation requirements & dependencies	
S6	If available, link to user manual - if formally published include a reference to the publication in the reference list	For example: *http* : *//mozart.github.io/documentation/*
S7	Support email for questions	

Table 4: Software metadata (optional)