

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, Avcılar, Istanbul, Turkey*

^c*Istanbul Technical University, Department of Computer Engineering, Sarıyer, 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 Indicates 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 $i = 1, 2, \dots, n - 1$, 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 corresponding 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 methods 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. $g_j(\cdot)$ 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	\dots	C_m
Weights	w_1	w_2	\dots	w_m
Functions	f_1	f_2	\dots	f_m
A_1	$g_1(A_1)$	$g_2(A_1)$	\dots	$g_m(S_A)$
A_2	$g_1(A_2)$	$g_2(A_2)$	\dots	$g_m(A_2)$
\vdots	\vdots	\vdots	\ddots	\vdots
A_n	$g_1(A_n)$	$g_2(A_n)$	\dots	$g_m(A_n)$

Table 2: Sample decision matrix

without loss of generality. When A_1, A_2, \dots, A_n are alternatives and C_1, C_2, \dots, 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 .

Multiple-criteria decision-making (MCDM) tools provide several algorithms for ordering or selecting alternatives and/or determining the weights when there is uncertainty. 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 commercial 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 im-](#)

49 [plementation details.](#)

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

```
54 abstract type MCDMResult end
```

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

```
57 struct TopsisResult <: MCDMResult
58     decisionMatrix::DataFrame
59     weights::Array{Float64,1}
60     normalizedDecisionMatrix::DataFrame
61     normalizedWeightedDecisionMatrix::DataFrame
62     bestIndex::Int64
63     scores::Array{Float64,1}
64 end
```

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

```
68 function topsis(
69     decisionMat::DataFrame,
70     weights::Array{Float64,1},
71     fns::Array{Function,1})::TopsisResult
```

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

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

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

79 and

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

82 present two distinct ways of install the package.

83 *2.2. Software Functionalities*

84 **Present the major functionalities of the software.**

85 The package implements methods for TOPSIS (Technique for Order Pref-
86 erence by Similarity to Ideal Solutions)(8), ELECTRE (Elimination and
87 Choice Translating Reality)(8), PROMETHEE (Preference Ranking Organi-
88 zation METHod for Enrichment of Evaluations)(8), DEMATEL (The Deci-
89 sion Making Trial and Evaluation Laboratory)(8), MOORA (Multi-Objective
90 Optimization By Ratio Analysis)(8), VIKOR (Vlsekriterijumska Optimizacija
91 I Kaompromisno Resenje in Serbian)(8), AHP (Analytic Hierarchy Process)(8),
92 GRA (Grey Relational Analysis)(8), NDS (Non-dominated Sorting)(8), SAW
93 (Simple Additive Weighting)(8), ARAS (Additive Ratio Assessment)(8), WPM

94 (Weighted Product Model)(8), WASPAS (Weighted Aggregated Sum Prod-
 95 uct ASsessment)(8), EDAS (Evaluation based on Distance from Average
 96 Solution)(8), MARCOS (Measurement Alternatives and Ranking according
 97 to COMpromise Solution)(8), MABAC (Multi-Attributive Border Approxi-
 98 mation area Comparison)(8), MAIRCA (Multi Attributive Ideal-Real Com-
 99 parative Analysis)(8), COPRAS (COMplex PROportional ASsessment)(?),
 100 COCOSO (Combined Compromise Solution)(8), and CRITIC (CRiteria Im-
 101 portance Through Intercriteria Correlation)(8) for multiple-criteria tools.
 102 The package also performs DEA for Data Envelopment Analysis(?) and
 103 includes a method for zero-sum game solver.

104 2.3. Sample code snippets analysis

105 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_1	4	90	100000	1500	2000
A_1	12	140	75000	550	1100

Table 3: Decision matrix

106 In this sample problem, a decision maker is subject to select an apartment
 107 by considering age of the building, size (in m^2 s), price (in \$), distance to city
 108 centre, and nearby population. The data can be entered as a two-dimensional
 109 array (matrix) or as a DataFrame object:

```

110 julia> using JMcDM, DataFrames
111 julia> df = DataFrame(
112   :age          => [6.0, 4, 12],
113   :size         => [140.0, 90, 140],
114   :price        => [150000.0, 100000, 75000],
115   :distance     => [950.0, 1500, 550],
116   :population   => [1500.0, 2000, 1100]);
117
118 julia> w = [0.35, 0.15, 0.25, 0.20, 0.05];
119 julia> fns = [minimum, maximum, minimum, minimum, maximum];
120 julia> result = topsis(df, w, fns);
121 julia> result.scores
122 3-element Array{Float64,1}:
123 0.5854753145549456
124 0.6517997936899308
125 0.41850223305822903
126
127 julia> result.bestIndex
128 2

```

129 The same analysis can be performed using `saw()` for the method of Simple
130 Additive Weighting

```

131 julia> result = saw(df, w, fns);
132 julia> result.bestIndex
133 2

```



```
julia> result1
```

Row	topsis	electre	cocoso	moora	vikor	wpm	waspas
	String	String	String	String	String	String	String
1			✓	✓	✓		
2	✓	✓				✓	✓
3		✓					

Figure 1: Image A

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

```
135 julia> result = wpm(df, w, fns);
```

```
136 julia> result.bestIndex
```

```
137 2
```

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

140 3. Illustrative Examples

141 Provide at least one illustrative example to demonstrate the major func-
 142 tions.

```
143 julia> methods1 = [:topsis, :electre, :vikor, :moora, :cocoso, :wpm, :waspas]
```

```
144 julia> result1 = summary(df, w, fns, methods1);
```

```
145 julia> methods2 = [:aras, :saw, :edas, :marcos, :mabac, :mairca, :grey];
```

```
146 julia> result2 = summary(df, w, fns, methods2);
```

```
[julia> result2
```

Row	grey	aras	saw	edas	marcos	mabac	mairca
	String	String	String	String	String	String	String
1							
2		✓	✓	✓	✓		
3	✓					✓	✓

Figure 2: Image A

147 4. Impact

148 **This is the main section of the article and the reviewers weight**
149 **the description here appropriately** Indicate in what way new research
150 questions can be pursued as a result of the software (if any). Indicate in
151 what way, and to what extent, the pursuit of existing research questions is
152 improved (if so). Indicate in what way the software has changed the daily
153 practice of its users (if so). Indicate how widespread the use of the software
154 is within and outside the intended user group. Indicate in what way the
155 software is used in commercial settings and/or how it led to the creation of
156 spin-off companies (if so).

157 *JMcDM* provides a moderate number of MCDA tools and utility func-
158 tions for developing new methods as well as performing decision analysis
159 using a single function call for each method. A researcher can easily perform
160 sequential analysis by changing the problem parameters and can compare
161 results of many tools. Existing software packages are mainly focused on pro-
162 viding a small subset of methods. *JMcDM* is an all-in-one solution and has
163 potential for increasing user productivity. Seeing the different results pro-

164 duced by the methods together also helps to discover which parameters the
165 research is more sensitive to and the reasons for them.

166 5. Conclusions

167 Set out the conclusion of this original software publication.

168 6. Conflict of Interest

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

172 Acknowledgements

173 The authors would like to thank the Editor-in-Chief, editors, reviewers for
174 providing extremely insightful comments, and other staff of the SoftwareX
175 Journal.

176 References

- 177 [1] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiob-
178 jective genetic algorithm: NSGA-II, IEEE Transactions on Evolutionary
179 Computation 6 (2) (2002) 182–197. doi:10.1109/4235.996017.
- 180 [2] S. Greco, J. Figueira, M. Ehrgott, Multiple criteria decision analysis,
181 Vol. 37, Springer, 2016.

- 182 [3] V. Yadav, S. Karmakar, P. P. Kalbar, A. Dikshit, PyTOPS: A python
183 based tool for TOPSIS, *SoftwareX* 9 (2019) 217–222. doi:10.1016/j.
184 softx.2019.02.004.
- 185 [4] W. Adams, R. Saaty, *Super decisions software guide* (2003).
186 URL https://superdecisions.com/sd_resources/v28_man01.pdf
- 187 [5] B. Mareschal, Y. D. Smet, Visual PROMETHEE: Developments of the
188 PROMETHEE - GAIA multicriteria decision aid methods, in: 2009 IEEE
189 International Conference on Industrial Engineering and Engineering Man-
190 agement, IEEE, 2009. doi:10.1109/ieem.2009.5373124.
- 191 [6] C. A. Bana e Costa, J.-M. de Corte, J.-C. Vansnick, MACBETH
192 (Measuring Attractiveness by a Categorical Based Evaluation Tech-
193 nique), American Cancer Society, 2011. doi:[https://doi.org/10.](https://doi.org/10.1002/9780470400531.eorms0970)
194 [1002/9780470400531.eorms0970](https://doi.org/10.1002/9780470400531.eorms0970).
- 195 [7] J. Bezanson, A. Edelman, S. Karpinski, V. B. Shah, Julia: A fresh
196 approach to numerical computing, *SIAM Review* 59 (1) (2017) 65–98.
197 doi:10.1137/141000671.
198 URL <https://doi.org/10.1137/141000671>
- 199 [8] C.-L. Hwang, K. Yoon, Methods for multiple attribute decision mak-
200 ing, in: *Multiple Attribute Decision Making*, Springer Berlin Heidelberg,
201 1981, pp. 58–191. doi:10.1007/978-3-642-48318-9_3.

202 **Current executable software version**

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

Nr.	(Executable) software meta-data 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: <i>https</i> : <i>//github.com/combogenomics/DuctApe/releases/tag/DuctApe-0.16.4</i>
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: <i>http</i> : <i>//mozart.github.io/documentation/</i>
S7	Support email for questions	

Table 4: Software metadata (optional)