

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 ?. 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 ?. 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	$g_m(A_1)$
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 .

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 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 ?. *Super Decisions* is a software package which is mainly focused on AHP (Analytic Hierarchy Process) and ANP (Analytic Network Process) ?. *Visual Promethee* implements Promethee method on Windows platforms ?. *M-BACBETH* is an other commercial software product that implements MACBETH with a easy to use GUI. ?. [List more software here if exist.]

JMcDM is designed to provide a developer-friendly library for solving multiple-criteria decision problems in Julia ?. 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

49 Give a short overview of the overall software architecture; provide a pic-
50 torial component overview or similar (if possible). If necessary provide im-
51 plementation details.

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

```
56   abstract type MCDMResult end
```

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

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

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

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

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

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

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

```
79 julia> using Pkg  
80 julia> Pkg.add("JMCDM")
```

81 and

```
82 julia> \]  
83 (@v1.5) pkg> add JMCDM
```

84 present two distinct ways of install the package.

85 *2.2. Software Functionalities*

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

87 The package implements methods for TOPSIS (Technique for Order Pref-
88 erence by Similarity to Ideal Solutions)?, ELECTRE (Elemination and Choice
89 Translating Reality)?, PROMETHEE (Preference Ranking Organization METHod
90 for Enrichment of Evaluations)?, DEMATEL (The Decision Making Trial
91 and Evaluation Laboratory)?, MOORA (Multi-Objective Optimization By
92 Ratio Analysis)?, VIKOR (VlseKriterijumska Optimizcija I Kaompromisno

93 Resenje in Serbian)?, AHP (Analytic Hierarchy Process)?, GRA (Grey Re-
 94 lational Analysis)?, NDS (Non-dominated Sorting)?, SAW (Simple Additive
 95 Weighting)?, ARAS (Additive Ratio Assessment)?, WPM (Weighted Prod-
 96 uct Model)?, WASPAS (Weighted Aggregated Sum Product ASsessment)?,
 97 EDAS (Evaluation based on Distance from Average Solution)?, MARCOS
 98 (Measurement Alternatives and Ranking according to COmpromise Solu-
 99 tion)?, MABAC (Multi-Attributive Border Approximation area Compari-
 100 son)?, MAIRCA (Multi Attributive Ideal-Real Comparative Analysis)?, CO-
 101 PRAS (COMplex PROportional ASsessment)?, COCOSO (Combined Com-
 102 promise Solution)?, and CRITIC (CRiteria Importance Through Intercrite-
 103 ria Correlation)? for multiple-criteria tools. The package also performs DEA
 104 for Data Envelopment Analysis? and includes a method for zero-sum game
 105 solver.

106 2.3. Sample code snippets analysis

107 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

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

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

```
112 julia> using JMCDM, DataFrames
113 julia> df = DataFrame(
114 :age          => [6.0, 4, 12],
115 :size         => [140.0, 90, 140],
116 :price        => [150000.0, 100000, 75000],
117 :distance     => [950.0, 1500, 550],
118 :population   => [1500.0, 2000, 1100]);
119
120 julia> w = [0.35, 0.15, 0.25, 0.20, 0.05];
121 julia> fns = [minimum, maximum, minimum, minimum, maximum];
122 julia> result = topsis(df, w, fns);
123 julia> result.scores
124 3-element Array{Float64,1}:
125 0.5854753145549456
126 0.6517997936899308
127 0.41850223305822903
128
129 julia> result.bestIndex
130 2
```

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

```
133 julia> result = saw(df, w, fns);
```



```
134 julia> result.bestIndex
```

```
135 2
```

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

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

```
138 julia> result.bestIndex
```

```
139 2
```

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

142 3. Illustrative Examples

143 Provide at least one illustrative example to demonstrate the major func-
144 tions.

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

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

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

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

149 4. Impact

150 This is the main section of the article and the reviewers weight
151 the description here appropriately Indicate in what way new research
152 questions can be pursued as a result of the software (if any). Indicate in
153 what way, and to what extent, the pursuit of existing research questions is

```
julia> result1
```

3x7 DataFrame

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

Figure 1: Image A

```
julia> result2
```

3x7 DataFrame

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

Figure 2: Image A

154 improved (if so). Indicate in what way the software has changed the daily
155 practice of its users (if so). Indicate how widespread the use of the software
156 is within and outside the intended user group. Indicate in what way the
157 software is used in commercial settings and/or how it led to the creation of
158 spin-off companies (if so).

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

168 5. Conclusions

169 Set out the conclusion of this original software publication.

170 6. Conflict of Interest

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

174 Acknowledgements

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

178 References

- 179 K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiob-
180 jective genetic algorithm: NSGA-II, IEEE Transactions on Evolutionary
181 Computation 6 (2) (2002) 182–197. doi:10.1109/4235.996017.
- 182 S. Greco, J. Figueira, M. Ehrgott, Multiple criteria decision analysis, Vol. 37,
183 Springer, 2016.
- 184 V. Yadav, S. Karmakar, P. P. Kalbar, A. Dikshit, PyTOPS: A python based
185 tool for TOPSIS, SoftwareX 9 (2019) 217–222. doi:10.1016/j.softx.
186 2019.02.004.
- 187 W. Adams, R. Saaty, Super decisions software guide (2003).
188 URL https://superdecisions.com/sd_resources/v28_man01.pdf
- 189 B. Mareschal, Y. D. Smet, Visual PROMETHEE: Developments of the
190 PROMETHEE - GAIA multicriteria decision aid methods, in: 2009 IEEE
191 International Conference on Industrial Engineering and Engineering Man-
192 agement, IEEE, 2009. doi:10.1109/ieem.2009.5373124.
- 193 C. A. Bana e Costa, J.-M. de Corte, J.-C. Vansnick, MACBETH (Measuring
194 Attractiveness by a Categorical Based Evaluation Technique), American

195 Cancer Society, 2011. doi:<https://doi.org/10.1002/9780470400531>.
196 eorms0970.

197 J. Bezanson, A. Edelman, S. Karpinski, V. B. Shah, Julia: A fresh approach
198 to numerical computing, SIAM Review 59 (1) (2017) 65–98. doi:10.1137/
199 141000671.
200 URL <https://doi.org/10.1137%2F141000671>

201 C.-L. Hwang, K. Yoon, Methods for multiple attribute decision making, in:
202 Multiple Attribute Decision Making, Springer Berlin Heidelberg, 1981, pp.
203 58–191. doi:10.1007/978-3-642-48318-9_3.

204 **Current executable software version**

205 Ancillary data table required for sub version of the executable software:
206 (x.1, x.2 etc.) kindly replace examples in right column with the correct
207 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)