SoftwareX 9 (2019) 217–222

Contents lists available at ScienceDirect

SoftwareX

journal homepage: www.elsevier.com/locate/softx

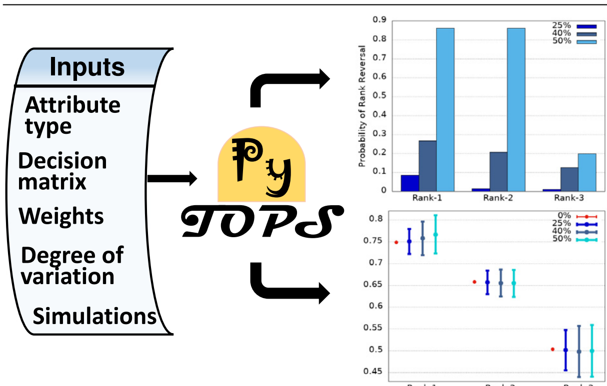
Original software publication

PyTOPS: A Python based tool for TOPSIS

Vinay Yadav [a], Subhankar Karmakar [a,b,c,*], Pradip P. Kalbar [b,c], A.K. Dikshit [a,c]

[a] Centre for Environmental Science and Engineering, Indian Institute of Technology Bombay, Mumbai 400076, India
[b] Inter Disciplinary Program on Climate Change, Indian Institute of Technology Bombay, Mumbai 400076, India
[c] Centre for Urban Science and Engineering, Indian Institute of Technology Bombay, Mumbai 400076, India

GRAPHICAL ABSTRACT



ARTICLE INFO

Article history:
Received 10 October 2018
Received in revised form 12 February 2019
Accepted 12 February 2019

Keywords:
Python
TOPSIS

ABSTRACT

The Technique for Order Preference by Similarity to Ideal Solution (TOPSIS) method determines the best solution from a set of alternatives with certain attributes. The best alternative is chosen based on its Euclidean distance from the ideal solution. TOPSIS is widely used in various multi-attribute decision making problems such as supply chain logistics, marketing management, environmental management or chemical engineering. Despite the extensive use of this method, there is no free and open-source software (FOSS) for TOPSIS with comprehensive post-analysis extensions. Therefore, this paper describes a Python-3 based tool PyTOPS for TOPSIS with the Berkeley Software Distribution (BSD)-3-Clause license.

* Corresponding author at: Centre for Environmental Science and Engineering, Indian Institute of Technology Bombay, Mumbai 400076, India.
E-mail address: skarmakar@iitb.ac.in (S. Karmakar).
URL: http://www.cese.iitb.ac.in/people/facinfo.php?id=skarmakar (S. Karmakar).

https://doi.org/10.1016/j.softx.2019.02.004

Code metadata

Current code version	v0.1
Permanent link to code/repository used for this code version	https://github.com/ElsevierSoftwareX/SOFTX_2018_196
Legal Code License	BSD-3-Clause
Code versioning system used	none
Software code languages, tools, and services used	Python3, PyQt5
Compilation requirements, operating environments & dependencies	numpy, sklearn, collections, math, xlrd, random, sys, QtCore, QtGui, QtWidgets
If available Link to developer documentation/manual	http://www.cese.iitb.ac.in/people/facinfo.php?id=skarmakar (Password: pytops2018)
Support email for questions	pytops-support@googlegroups.com

Software metadata

Current software version	v0.1
Permanent link to executables of this version	uploaded with manuscript
Legal Software License	BSD-3-Clause
Computing platforms/Operating Systems	Linux, Microsoft Windows
Installation requirements & dependencies	numpy, sklearn, collections, math, xlrd, random, sys, PyQt5 (for linux only)
If available, link to user manual - if formally published include a reference to the publication in the reference list	http://www.cese.iitb.ac.in/people/facinfo.php?id=skarmakar (Password: pytops2018)
Support email for questions	pytops-support@googlegroups.com

1. Motivation and significance

Multi-attribute decision making (MADM) is one of the renowned branches of decision making. This branch deals with the development of computational and mathematical tools for subjective evaluation of given alternatives with respect to the identified attributes [1]. Alternatives are different choices of actions present for the decision maker, which are prioritized and ranked. Many MADM approaches have been developed for the ranking of alternatives such as the analytic hierarchy process (AHP), weighted sum model (WSM), simple additive weighting (SAW), linear programming technique for multidimensional analysis of preference (LINMAP), multi-objective optimization on the basis of ratio analysis (MOORA), ELimination Et Choix Traduisant la REalité (ELECTRE) and Technique for Order Preference by Similarity to Ideal Solution (TOPSIS) [2,3]. An extensive list of available software for these different MADM approaches is provided on the official web-page of international society on multiple criteria decision making [4]. Among the developed approaches, TOPSIS is utilized in the diverse fields of research [5–7]. TOPSIS chooses the best alternative based on the shortest and farthest Euclidean distances from the positive ideal solution and negative ideal solution, respectively [8–10]. The algorithm of TOPSIS method is summarized using pseudo-code format in Algorithm 1.

TOPSIS was developed by Hwang and Yoon in 1981, and has been used in various MADM problems such as supply chain logistics, marketing management, environmental management or chemical engineering [11,12]. TOPSIS is preferred over other approaches because of (i) its suitability for large number of attributes and alternatives; (ii) requirement of limited subjective inputs; (iii) its logical and programmable behavior; and (iv) comparative consistency in the alternative ranking [13]. Despite the extensive use of TOPSIS in various fields of study, there is no free and open-source software (FOSS) available for TOPSIS, especially with comprehensive post-analysis extensions, to the best of authors' knowledge. Therefore, this paper describes a Python-3 based tool PyTOPS for TOPSIS with the Berkeley Software Distribution (BSD)-3-Clause license for Microsoft Windows and Linux platforms.

PyTOPS is designed to provide a user-friendly graphical interface to solve large MADM problems using TOPSIS. PyTOPS is a simple but powerful package for post-analysis of the results

Algorithm 1 Steps of TOPSIS method in pseudo-code format

1: Construction of normalized decision matrix *i.e.*,

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^{m} x_{ij}^2}} \quad \forall j$$

where r_{ij} and x_{ij} are the elements of normalized and original decision matrix respectively

2: Construction of weighted normalized decision matrix *i.e.*,

$$v_{ij} = r_{ij} * w_j \quad \forall i, j$$

where w_j is the assigned weight to attribute j

3: Determination of ideal ($\mathcal{A}^+$) and negative-ideal ($\mathcal{A}^-$) solutions *i.e.*,

$$\mathcal{A}^+ = \{(\max_j v_{ij} | i \in I), (\min_j v_{ij} | i \in I'); \forall j\} = \{v_1^+, v_2^+, \ldots\}$$

$$\mathcal{A}^- = \{(\min_j v_{ij} | i \in I), (\max_j v_{ij} | i \in I'); \forall j\} = \{v_1^-, v_2^-, \ldots\}$$

where I and I' are associated with benefit and cost attributes respectively.

4: Calculation of separation measure *i.e.*,

$$S_i^+ = \sqrt{\sum_{i=1}^{n} (v_{ij} - v_i^+)^2} \quad \forall j$$

$$S_i^- = \sqrt{\sum_{i=1}^{n} (v_{ij} - v_i^-)^2} \quad \forall j$$

5: Calculation of relative closeness to the ideal solution *i.e.*,

$$C_j^+ = \frac{S_j^+}{S_j^+ + S_j^-}$$

6: Ranking of alternatives based on C_j^+ values such as probability of rank reversibility, the mean and standard deviation of relative closeness to the ideal solutions. The mean and standard deviation of the relative closeness concerning each alternative provide the liberty to belittle some alternative with an intervention of human understanding. The influence of some attributes with their given weights can also be examined with multiple simulations using PyTOPS. These advanced applications will facilitate many researchers and decision makers to utilize enhanced computational capability for studying TOPSIS method.

Many software packages have been developed for solving MADM problems using TOPSIS such as DECERNS [14], Triptych: TOPSIS [15], SANNA [16], Topsis Solver 2012 [17] and TechSelect [18]. DECERNS is a web-based and desktop decision support system for MADM models with a wide range of applications in scientific and practical problems [19]. However, it does not have any post analysis functionality and is distributed with a proprietary commercial license. The Triptych package for TOPSIS analysis is developed and distributed by Statistical Design Institute with proprietary commercial license. This software requires Microsoft Excel and limits the number of attributes and alternatives to 200. SANNA and Topsis Solver 2012 also need Microsoft Excel to function and limits the number of attributes and alternatives to 50 and 20, respectively. TechSelect is a scenario-based decision support tool with TOPSIS nested on Microsoft Excel. Evidently, the available software are either commercial or depend on some other commercial package such as Microsoft Excel and have limitations in terms of the attributes and alternatives in the decision matrix. The goal of PyTOPS is to provide a FOSS equivalent for TOPSIS with a large decision matrix (thousands of attributes and alternatives) and up to a million simulations for post-analysis.

2. Software description

The standard steps for the TOPSIS method are (i) formation of normalized decision matrix; (ii) weighted normalized decision matrix formation; (iii) ideal and negative-ideal solution determination; (iv) separation measure calculation; (v) calculation of relative closeness to ideal solutions; and (vi) ranking of preference order. The normalization of the decision matrix is carried out using inbuilt *sklearn.preprocessing* package followed by a matrix multiplication with the weights to construct weighted normalized decision matrix. Ideal and negative ideal solutions are identified based on predetermined cost and benefit attributes. Separation measure and relative closeness are calculated by defining specific functions respectively. Alternatives are then ranked based on relative closeness to the ideal solution using various inbuilt functions of *numpy* package. The subsequent sections explain these functionalities in detail.

2.1. Software architecture and functionalities

As can be seen in Fig. 1, PyTOPS has two blocks: Inputs and Outputs. The Inputs block lists five components: (i) attribute type; (ii) upload decision matrix; (iii) upload weights; (iv) degree of variation in weights; and (v) number of simulations. The input attribute type component identifies cost and benefit based on the binary values (0 for benefit and 1 for cost) given by the user. These binary values are then appended to the columns of decision matrix. This practice ushers defined functions to recognize attributes of different types, which are used in the determination of the ideal and negative-ideal solutions. *OrderedDict* subclass is used to remember the attribute type (given in *list* form) for further operations. Components two and three have pushbuttons for uploading decision matrix and attribute weights with *xlsx* extensions, respectively. *xlsx* is the most common file extension for spreadsheets and can be exported using many free software such as *LibreOffice Calc* and *Google Sheets*. The data of these files are extracted using *xlrd* library. The weights to attributes are given by decision makers based on the relative importance and are generally subjective in nature. This subjectivity is always associated with inherent uncertainty. To address this issue, the degree of variation component is added with double spin box. This double spin box has a variation up to 1.00 (1.00 means decision maker is willing to have a variation of 100% in provided weights). To analyze the influence of this variation multiple simulations may be carried out in the next component using a spin box. Number of simulations can be increased to one million.

Outputs block contains four components: (i) rank with varying weights; (ii) probability of rank reversal; (iii) mean of relative closeness to ideal solution; and (iv) standard deviation of relative closeness to the ideal solution. After uploading all necessary files and other information in Inputs block, next step is to run the script with RUN pushbutton. RUN button fills remaining text boxes of Outputs block. The first component gives respective ranks of alternatives with the chosen weights from the user defines variation in weights (see component 4 of Inputs). This text box first prints the weights and subsequently prints the ranks in a loop configuration. MADM approaches have a prominent problem of rank reversal *i.e.*, change in the rank ordering of preferable alternatives. For the quantification of this problem, the probability of rank reversal for all the alternatives in two-column format is printed in the second component. This probability palpably reflects the robustness of alternatives ranks. The third and fourth components print mean and standard deviation of relative closeness to the ideal solution in the respective text boxes for every alternative. The mean and standard deviations values bestow an opportunity to the decision maker for choosing an inferior, but best with human perceptibility, alternative. All printed values in each text boxes of all components can be saved by using save tool button with *txt* extension. The overall structure of PyTOPS is given in Fig. 2.

2.2. Sample code snippets analysis

Source code of PyTOPS are written in Python3. The codes may be further modified for developing a tool for fuzzy TOPSIS methods. Existing functions such as *normalize* of *sklearn.preprocessing* and *xlrd* are operational only for deterministic values (see the snippet below). The possible modification in these utility functions will allow decision makers to choose a preferable alternative under fuzzy uncertainty.

```
workbook = xlrd.open_workbook('decision_matrix.xlsx')
sheet1 = workbook.sheet_by_index(0)
_matrix=[]
for row in range (sheet1.nrows):
    _row = []
    for col in range (sheet1.ncols):
        _row.append(sheet1.cell_value(row,col))
    _matrix.append(_row)
A = np.matrix(_matrix)
from sklearn.preprocessing import normalize
B = normalize(A, norm='l2', axis=0)
```

3. Illustrative example

The working of PyTOPS is illustrated using Example 1, which a commonly practiced simple numerical problem.

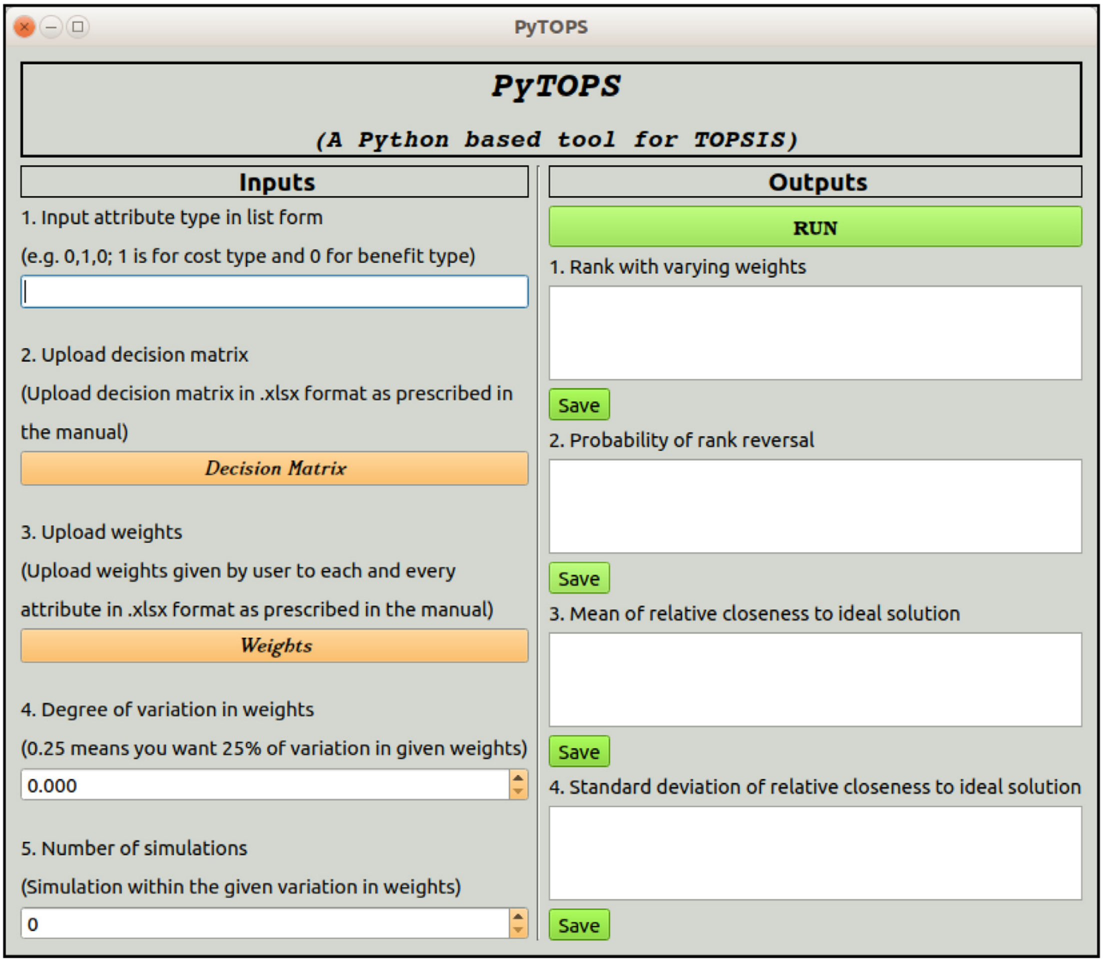


Fig. 1. Graphical user interface of PyTOPS.

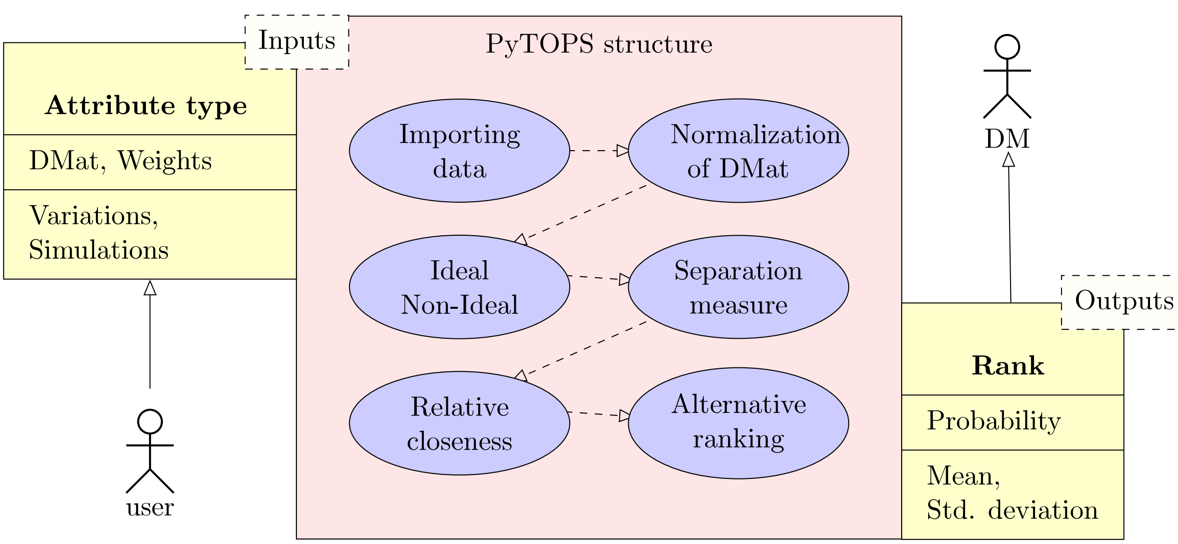


Fig. 2. Unified modeling language diagram to show the structure of PyTOPS; where DMat is decision matrix, Std deviation is standard deviation and DM is decision maker; Inputs and Outputs are same as given in Fig. 1.

Example 1. A manager of an international firm is trying to evaluate the requirement of a new facility for the replacement of current structure with A_1, A_2, A_3 and A_4 as alternatives. (S)he considers three attributes i.e. durability, reliability and capability for this evaluation. (S)he also assign certain weights for each of these attributes as shown in Table 1.

As can be seen in Table 1, all of the attributes in Example 1 are benefit type. Therefore, the first step is to provide 0, 0, 0 in the first component (*i.e.* Input attribute type) of Inputs block.

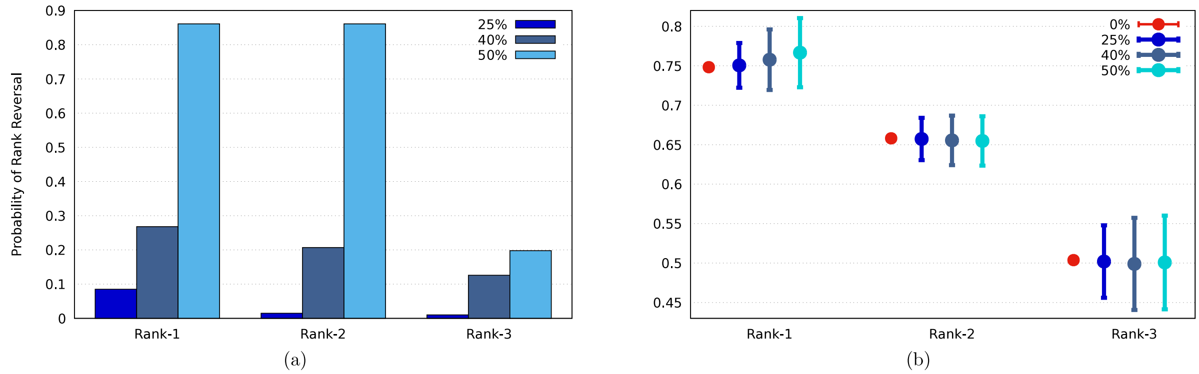


Fig. 3. (a) Probability of rank reversal for top ranks; and (b) Mean and standard deviation of closeness to the ideal solution for the top ranks of alternatives (*i.e.*, A_2, A_3 and A_4).

Table 1
Alternatives and attributes for Example 1 [20].

	Durability	Reliability	Capability
Weights	0.3	0.4	0.3
A_1	5	4	8
A_2	7	8	6
A_3	8	6	8
A_4	7	6	4

The decision matrix and assigned weights are then uploaded using respective push buttons (see supplementary information for *xlsx* files). PyTOPS ranks the alternatives as $A_2 > A_3 > A_4 > A_1$. These results are intuitive as A_2 has the highest score in the Reliability attribute, which is the highest weighted attribute. For the demonstration purpose, a variation of 25%, 40% and 50% in the assigned weights have been considered. In all these variations one thousand simulations are performed. Fig. 3 (a) depicts the probability of rank reversibility under the given variations of weights (*i.e.*, 25%, 40% and 50%) graphically. Fig. 3(b) gives means and standard deviations of the closeness to the ideal solution of top alternatives (*i.e.*, A_2, A_3 and A_4).

4. Impact

Most of the MADM methods principally depend on the subjective judgments of decision makers [21]. TOPSIS method also requires weights associated with attributes as subjective information [22]. This subjectivity brings uncertainty into the outcomes. PyTOPS provides liberty for decision makers to analyze how uncertainty of outcomes is apportioned to the subjectivity of input weights by allowing multiple simulations with uniformly distributed values around the given information. Definiteness of preferred alternatives can be examined using this functionality without creating new scenarios. This functionality also focuses on the robustness testing of the decision model *e.g.*, if the best alternative changes with a very minute perturbation in weight, then decision makers may question the robustness of the model. On the other hand, models are considered robust if a comparatively larger perturbation of weights does not change the rank of alternatives. Conclusively, PyTOPS has the capability for robustness examination and uncertainty analysis of decision models.

PyTOPS has been used to solve a two stage multi-attribute decision making model for selecting appropriate locations of waste transfer stations in the city of Nashik (India) reported in [23,24]. This model chooses best locations of municipal solid waste management facilities in the urban centers based on the attributes identified by all level of stake holders. PyTOPS provides the rank to all the possible combinations of these facilities along with the probability of their reversal. Moreover, it also gives the mean and standard deviation of the relative closeness in multiple simulations, which is an important reckon for municipal officials.

The use of PyTOPS is not restricted to academicians and may be used by any decision maker in industries and governments under BSD license as MADM problems extensively encounters in these domains. The examples of many MADM problems in industries and governments can be found in literature (for example, see [25,26]). PyTOPS may also be used in educational institutes in operation research courses as it has a very user-friendly graphical interface. Additionally, the present software may be of interest to decision makers who works with real large data sets with thousands of alternatives and attributes.

5. Conclusions

This paper presents a Python based tool PyTOPS for the MADM method TOPSIS with a wide range of post-analysis extensions such as probability of rank reversibility, the mean and standard deviation of relative closeness to the ideal solution. All the functionalities of PyTOPS are explained using a numerical example and all necessary files to solve the example problem are provided as supplementary material. The use of state of the art packages bestows a high efficiency which makes PyTOPS a suitable tool for large datasets and concurrently reduces the processing time. Users from diverse backgrounds are invited to use PyTOPS and to provide feedback, which will be taken forward to improve the functionality in later versions. Apart from routine maintenance and bug fixation, future study on PyTOPS will incorporate fuzzy uncertainty of decision models as explained in Section 2.2.

Acknowledgments

The authors would like to thank the Editor-in-Chief and anonymous Reviewers for providing extremely insightful comments, which has improved the quality of this manuscript. VY thanks Ministry of Human Resource and Development, Government of India for providing his Ph.D. fellowship and IIT Bombay for computational support. VY also acknowledges Mr. Ranjeet Kumar Singh and Mr. Pratiman Patel for their help.

Conflict of interest

The authors declare that there is no conflict of interest.

References

[1] Zavadskas EK, Turskis Z, Kildienė S. State of art surveys of overviews on MCDM/MADM methods. Technol Econ Dev Econ 2014;20(1):165–79.

[2] Mardani A, Jusoh A, MD Nor K, Khalifah Z, Zakwan N, Valipour A. Multiple criteria decision-making techniques and their applications–a review of the literature from 2000 to 2014. Econ Res 2015;28(1):516–71.

[3] Zavadskas EK, Govindan K, Antucheviciene J, Turskis Z. Hybrid multiple criteria decision-making methods: A review of applications for sustainability issues. Econ Res 2016;29(1):857–87.

[4] International Society on MCDM. Software Related to MCDM, 2019. https://www.mcdmsociety.org/content/software-related-mcdm. [Online; Accessed 09 January 2019].

[5] Zavadskas EK, Sušinskas S, Daniūnas A, Turskis Z, Sivilevičius H. Multiple criteria selection of pile-column construction technology. J Civ Eng Manag 2012;18(6):834–42.

[6] Zavadskas EK, Turskis Z, Volvačiovas R, Kildiene S. Multi-criteria assessment model of technologies. Stud Inf Control 2013;22(4):249–58.

[7] Zavadskas EK, Turskis Z, Bagočius V. Multi-criteria selection of a deep-water port in the Eastern Baltic Sea. Appl Soft Comput 2015;26:180–92.

[8] Lai YJ, Liu TY, Hwang CL. TOPSIS for MODM. European J Oper Res 1994;76(3):486–500.

[9] Zavadskas EK, Ustinovičius L, Turskis Z, Peldschus F, Messing D. LEVI 3.0-multiple criteria evaluation program for construction solutions. J Civ Eng Manag 2002;8(3):184–91.

[10] Šaparauskas J, Turskis Z. Evaluation of construction sustainability by multiple criteria methods. Technol Econ Dev Econ 2006;12(4):321–6.

[11] Behzadian M, Otaghsara SK, Yazdani M, Ignatius J. A state-of the-art survey of TOPSIS applications. Expert Syst Appl 2012;39(17):13051–69.

[12] Yoon K, Hwang CL. Multiple attribute decision making: methods and applications. Springer-Verlag Berlin; 1981.

[13] Kalbar PP, Karmakar S, Asolekar SR. Selection of an appropriate wastewater treatment technology: A scenario-based multiple-attribute decision-making approach. J Environ Manag 2012;113:158–69.

[14] Yatsalo B, Didenko V, Tkachuk A, Gritsyuk S, Mirzeabasov O, Slipenkaya V, Babutski A, Pichugina I, Sullivan T, Linkov I. Multi-criteria spatial decision support system DECERNS: application to land use planning. In: Societal impacts on information systems development and applications. IGI Global; 2012.

[15] Statistical Design Institute. Triptych: TOPSIS Goal-based multi-criteria decision making. 2017. http://aiweb.techfak.uni-bielefeld.de. [Online; Accessed 09 December 2017].

[16] Jablonsky J, Urban P. MS Excel Based system for multicriteria evaluation of alternatives. University of Economics Prague 1998;1–8.

[17] Amari T. Topsis Solver 2012. 2017. http://www.tahlil-amari.com/. [Online; Accessed 09 December 2017].

[18] Kalbar PP, Karmakar S, Asolekar SR. Life cycle-based decision support tool for selection of wastewater treatment alternatives. J Cleaner Prod 2016;117:64–72.

[19] Yatsalo B, Didenko V, Gritsyuk S, Sullivan T. DECERNS: A framework for multi-criteria decision analysis. Int J Comput Intell Syst 2015;8(3):467–89.

[20] Tzeng GH, Huang JJ. Multiple attribute decision making: methods and applications. CRC press; 2011.

[21] Yang GL, Yang JB, Xu DL, Khoveyni M. A three-stage hybrid approach for weight assignment in MADM. Omega 2017;71:93–105.

[22] Ishizaka A, Nemery P. Multi-criteria decision analysis: methods and software. John Wiley & Sons; 2013.

[23] Yadav V, Kalbar PP, Karmakar S, Dikshit A. A two stage multi-attribute decision making model for selecting appropriate locations of waste transfer stations in urban centers. 2019, (Under Review).

[24] Yadav V, Karmakar S, Dikshit A, Bhurjee A. Interval-valued facility location model: an appraisal of municipal solid waste management system. J. Cleaner Prod. 2018;171:250–63.

[25] Sun LY, Miao CL, Yang L. Ecological-economic efficiency evaluation of green technology innovation in strategic emerging industries based on entropy weighted TOPSIS method. Ecol Indic 2017;73:554–8.

[26] Bilbao Terol A, Arenas Parra M, Cañal Fernández V, Antomil Ibias J. Using TOPSIS for assessing the sustainability of government bond funds. Omega 2014;49:1–17.