Fuzzy vs. Crisp in Uncertainty-aware Service Selection: Enabling Sustainability on Multimedia Event Processing

Felipe Arruda Pontes
Insight Centre for Data Analytics
University of Galway
Galway, Ireland
0000-0002-9062-7653

Michael Schukat
School of Computer Science
University of Galway
Galway, Ireland
0000-0002-6908-6100

Edward Curry
Insight Centre for Data Analytics
University of Galway
Galway, Ireland
0000-0001-8236-6433

Abstract*—The high energy consumption projections of Cloud/Edge applications urge the development of ecologically sustainable Multimedia Event Processing (MEP) systems. In these applications, a Multi-Criteria Decision Making (MCDM) problem must be solved when selecting the best service workers alternatives for processing user queries, according to the user-specific performance criteria requirements of energy, accuracy and speed. Moreover, fuzzy logic provides a well-established method, such as the fuzzy TOPSIS, for dealing with the uncertainties arising from real-world scenarios, where ambiguities of user requirement interpretations and imprecision of measurement of the computing devices may directly impact this decision-making process. However, this fuzzy method is more complex and computationally intensive than the original (crisp) TOPSIS. Therefore, it is crucial to understand to what degree the fuzzy and crisp methods may be used interchangeably and still get the same results to avoid unnecessary complexity in sustainable MEP solutions in a real-world context. In our work, we developed a fuzzy TOPSIS ranking method for handling the uncertainties of the user criteria weights and service worker alternative ratings. Contrary to a previous study, we provide evidence that replacing the fuzzy TOPSIS method with its crisp counterpart significantly affects the ranking results when applied to a real-world scenario, with contradiction rates higher than 60% in most scenarios explored, which suggests that it is not viable to interchange these methods without consequences to the sustainability efforts of an MEP application.

I. Introduction

Over the last few years, the advance of Big Data applications and the Internet of Things have highly impacted the environment [1]. In 2016 the US energy consumption of data centres was about 1.8% of the whole country [2]. For the next decade, the combined energy consumption for data centres, IoT and other devices is estimated to contribute to over 14% of worldwide energy consumption [3]. Meanwhile, Multimedia Event Processing (MEP) applications deployed on the Cloud and Edge, which have been applied to a wide range of problems [4], [5], generally make use of State-of-the-Art Deep Neural Network (DNN) models for processing the multimedia data [5]. Moreover, some DNN model operations can have high energy consumption [6], emphasising the need for sustainable MEP applications on the Cloud and Edge. Moreover, to adapt to the constant changes in the deployment environment and changes in user performance requirements, these MEP applications usually implement some form of self-adaptive control. With these adaptations, the system is presented with a Multi-Criteria Decision Making (MCDM) problem, in which the service worker profiles in the network must be ranked and selected according to the end-user's performance criteria requirements (i.e., energy consumption, accuracy, and latency), so that each user's query processing is done by the best-ranked worker available for the task required by that user. This decision-making process can be directly impacted by uncertainties arising from different interpretations of the user performance requirements and imprecision in the measurements of the service worker's devices [7]–[9]. Meanwhile, fuzzy logic has long been proven as a well-established method for handling imprecision and ambiguities in MCDM problems [10]. The TOPSIS (Technique for Order Performance by Similarity to Ideal Solution) [11] is a commonly used MCDM ranking method, which was later extended to the fuzzy domain by Chen to make use of fuzzy logic to handle some of these uncertainties [10]. However, this fuzzy uncertainty awareness may present an overhead in complexity compared to the original (crisp) TOPSIS [12]. Therefore, to avoid unnecessary complexity, it is crucial to understand in what cases one can make use of a simpler crisp TOPSIS instead of the more complex fuzzy method and still get the same ranking results, in particular when using real-world data in the context of ecologically sustainable MEP applications on the Cloud and Edge.

This situation elicits the research question: *"What is the expected rate of contradictions in the ranking results when replacing an uncertainty-aware fuzzy TOPSIS method with its uncertainty-naive crisp counterpart, when considering both benefit and cost criteria types covered by these methods in*

This publication has emanated from research conducted with the financial support of Science Foundation Ireland (SFI) under Grant Number SFI/12/RC/2289_P2, co-funded by the European Regional Development Fund.

a real-world scenario of sustainable MEP on the Cloud and Edge?".

As a response to this question, our paper provides the following contributions:

- **Development of an uncertainty-aware service worker selector for sustainable MEP over the Cloud and Edge.** This selector uses the fuzzy TOPSIS method to rank the available service worker alternatives while handling the ambiguities of the linguistic variable related to the user-defined criteria weights and the hardware measurements imprecision in the service worker alternatives, which can impact the sustainability goals of the system;
- **Provide evidence, contrary to a previous study [12], that replacing the fuzzy TOPSIS method with its crisp counterpart significantly affects the ranking results when applied to a real-world scenario.** Our comparative results show contradiction rates in the ranking results of more than 60% in most scenarios when using real-world data inputs, full coverage of the TOPSIS using both benefit and cost criteria, multiple scenarios for the user criteria requirements (i.e., energy, speed and accuracy), and a range of alternatives on different deployment environments setups.

The remainder of this paper is organised as follows. Section II describes the related works and the research gap targeted by our study. Next, in Section III, we describe the use of the TOPSIS method in our study case of sustainable MEP applications. In Section IV, we give a brief description of the crisp and fuzzy TOPSIS methods. Section V presents the experiments for comparing both methods. Section VI discusses the results of the fuzzy and crisp TOPSIS comparison. Finally, in Section VII, we draw conclusions and present future works.

II. Related Works

This section will briefly describe the related works on service selection on fuzzy uncertainty-aware Sustainable Multimedia Event Processing frameworks for Cloud and Edge. In RobusT2Scale [13], an uncertainty-aware self-adaptive Cloud elastic scaling solution is proposed for balancing response time and computational resources, using Fuzzy Q-Learning for knowledge evolution and a rule-based fuzzy controller for handling the uncertainty in the system administrator (sys-admin) policies definitions and system monitoring. Similarly, [7] proposes a proof-of-concept rule-based fuzzy controller for uncertainty-aware self-adaptive scaling of service workers with a focus on resource-constrained Edge deployment according to the sys-admin requirements. [8] presents a rule-based fuzzy self-adaptive demand- and uncertainty-aware task management that maximises performance and reduces energy consumption by analysing the decision of offloading tasks from Edge nodes to their peers in a Cloud and Edge web service study case. [14] uses the fuzzy TOPSIS method for web service selection based on the Quality of Service (QoS) definitions while considering benefit and cost criteria in their use case of a diabetes monitoring system, but focusing on HTTP-based connections instead of the multimedia data streams inherent to MEP systems. [9] describes how the scheduler interpretability is strongly affected in both fuzzy and crisp scheduling solutions for cloud computing applications with a focus on the economy and energy consumption impact of adaptive and uncertainty-aware schedulers.

Additionally, an overview of fuzzy MCDM methods for service selection can be seen in the comprehensive review done by [15]. [16] has explored the comparison of fuzzy MCDM ranking methods. Moreover, comparing fuzzy methods and their crisp counterpart has also been studied. [17] shows a high disparity of ranking results from the fuzzy and crisp VIKOR MCDM methods when comparing their rankings using data from randomly generated MCDM problems. On the other hand, [12] points out in their study that the fuzzy TOPSIS and its crisp counterpart have little difference in their ranking. These last two works, however, do not utilise real-world input data nor consider the impact of the uncertainties on the interpretation of the criteria weights and on the cost functions present in these methods (i.e., no cost criteria such as energy consumption is used). Moreover, none of the related works considers the ambiguities in interpreting end-user-specific performance criteria requirements for the system.

In a real-world context, there are dynamic and complex environments dependent on variables with vague and ambiguous human definitions and imprecise measurements [10], [18], [19]. Therefore, when comparing TOPSIS methods that are aware or naive to these uncertainties, a more realistic approach may be to use real-world data measurements instead of randomised data. It is also essential to consider the full extent of the TOPSIS method by considering benefit and cost criteria when comparing the fuzzy and crisp versions.

Overall, there is a gap in current research on the difference in ranking results when using the crisp TOPSIS compared to the fuzzy TOPSIS method, which covers both the benefit and cost criteria and takes into consideration the ambiguities of the user-defined performance criteria requirements in a real-world scenario such as sustainable MEP applications on the Cloud and Edge.

III. Application of Fuzzy TOPSIS for Service Selection

In the context of sustainable MEP applications deployed on the Cloud and Edge, the fuzzy TOPSIS method can be used to better select the service workers to process a user query according to the user performance criteria requirements of energy, accuracy and speed (i.e., latency). Figure 1 depicts an example of an MEP application for processing video streams from a traffic management system. In this case, a user queries the system for cars with invalid license plates. The traffic camera images are processed using a service with multiple workers deployed on both Cloud and Edge environments, using pre-trained DNN models to detect invalid license plates. The fuzzy TOPSIS is used to rank the best service workers (i.e., Edge Worker B) from the available alternatives based on their ratings of energy, accuracy and speed and according to the importance of the criteria weights as defined in the

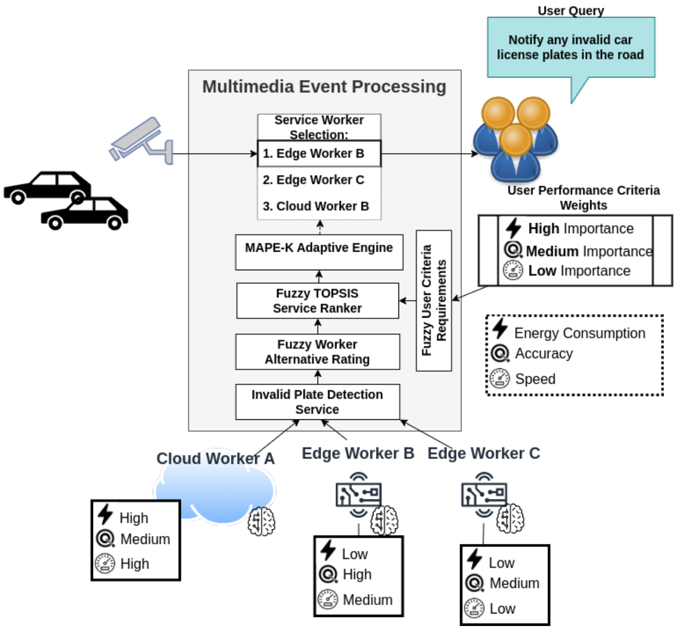


Fig. 1: Example of fuzzy service selection in a DNN-based MEP system for traffic management deployed on the Cloud and Edge. The best-ranked service worker (Edge Worker B) is selected from the available alternatives for processing the user query based on the user's performance criteria requirements.

user query (i.e., High, Medium, and Low importance for energy, accuracy and speed, respectively). Once ranked, the best alternatives are selected during an adaptation cycle. These adaptations can be triggered by changes in the deployment environment (e.g., when a worker is no longer available) or in the current user requirements (e.g., the inclusion of new queries with different performance requirements).

IV. Background of the TOPSIS Method

The TOPSIS method, as proposed initially by Hwang and Yoon [11], focuses on solving crisp MCDM problems by prioritising the alternatives that are closer (in terms of euclidean distance) to the Positive Ideal Solution (PIS) and farthest from the Negative Ideal Solution (NIS). This distance is the solution that minimises the cost criteria and maximises the benefit criteria. The input of this method is composed of two parts, the decision matrix D and weights vector W:

$$D = \begin{array}{c} A_1 \\ A_2 \\ A_i \\ \dots \\ A_m \\ \\ \end{array} \begin{array}{c} \begin{bmatrix} X_{11} & X_{12} & X_{1j} & \dots & X_{1n} \\ X_{21} & X_{22} & X_{2j} & \dots & X_{2n} \\ X_{i1} & X_{i2} & X_{ij} & \dots & X_{in} \\ \dots & \dots & \dots & \dots & \dots \\ X_{m1} & X_{m2} & X_{mj} & \dots & X_{mn} \end{bmatrix} \\ \begin{array}{ccccc} C_1 & C_2 & C_j & \dots & C_n \end{array} \end{array} \tag{1}$$

$$W = \begin{bmatrix} w_1 & w_2 & w_j & \dots & w_n \end{bmatrix} \tag{2}$$

The decision matrix D contains the values X_{ij} of each alternative A_i for every criterion C_j, over m alternatives and n criteria. Meanwhile, the weights vector W, contains the weights of importance of each criterion C_j, as defined by the decision maker.

The general steps for the TOPSIS method are defined in Algorithm 1.

TABLE I: Linguistic Terms for the Criteria Weight Importance and Alternative Rating based on [10]

Linguistic Terms	**Triangular Fuzzy Number**	
	Criteria Weights	**Alternative Ratings**
High	(0.7, 0.9, 1.0)	(7, 9, 10)
Medium-High	(0.5, 0.7, 0.9)	(5, 7, 9)
Medium	(0.3, 0.5, 0.7)	(3, 5, 7)
Medium-Low	(0.1, 0.3, 0.5)	(1, 3, 5)
Low	(0.1, 0.1, 0.3)	(1, 1, 3)

Algorithm 1 TOPSIS General Steps

Input: D, W ▷ *As defined in Eq. 1 and 2*
Output: $Rank$, CC ▷ *Outputs the rank and Closeness Coefficients*

1: Create a normalised decision matrix R
2: Create the weighted normalised decision matrix V
3: Calculate PIS (A^*) and NIS (A^-)
4: Calculate the distance measures d_i^* and d_i^- for benefit and cost criteria, respectively.
5: Calculate Closeness Coefficient for each alternative (CC_i).
6: Define $Rank$ order according to each CC_i.
7: **return** $Rank$, CC

In this paper, the TOPSIS method was implemented in Python using the Scikit-Criteria toolkit, which uses a scale conversion based on the total sum of values for the criteria weights (**Step 2**) and a vector normalisation for the decision matrix (**Step 1**) [20].

A. Fuzzy TOPSIS Extension

Chen created the first fuzzy extension of the TOPSIS method [10]. In this extension, three extra steps of pre-processing are necessary before following the rest of the TOPSIS algorithm:

- **F.1**: Define the Linguistic Variables Scales
- **F.2**: Apply Criteria/Alternatives Scale Conversion
- **F.3**: Aggregate criteria weights and alternatives ratings

In **F.1**, the Linguistic Variables for the criteria weights and alternative ratings are defined using a 10-points conversion scale [10]. To avoid division-by-zero issues, the range of values on this scale was set from 1 to 10 (or 0.1 to 1.0). These values are expressed using triangular fuzzy numbers (TFN) to represent five linguistic terms, as shown in Table I.

Next, in **F.2**, each alternative's values are converted using the previously defined 10-point scale [16]. A corresponding TFN is used by selecting the one with the closest modal value to the 10-point scale. That is, three alternatives with the values of 10, 50 and 100 as a criterion would translate into the respective linguistic terms of "*Low*", "*Medium*", and "*High*", with a corresponding TFN of "(1, 1, 3)", "(3, 5, 7)", "(7, 9, 10)", respectively. Finally, the aggregation methods from step **F.3** used in this paper are the same used in [10], in which the criteria and alternative ratings of all decision makers are averaged for their values on the left, modal and right of their TFN. Similarly, the rest of the general TOPSIS steps (1 through 6) for the fuzzy extension are followed according to what was defined by Chen [10]. Our fuzzy TOPSIS method was implemented in Python, using the Numpy library and validated with multiple tests against the example case presented in Chen [10].

B. Pre-Scaled Crisp TOPSIS

To better contrast the results from previous fuzzy and crisp TOPSIS comparisons [12], we also consider a slightly modified version of the crisp TOPSIS with the application of a 10-points conversion scale onto the alternative ratings, as would be done in step **F.2** of the fuzzy TOPSIS. For example, suppose three alternatives with the "energy consumption" criterion of 10, 50 and 100 Watts. A rating of 1, 5, and 10 would be used in this pre-scaled crisp TOPSIS version instead of directly using their values in Watts, as would be done with the regular crisp TOPSIS.

V. Fuzzy vs Crisp TOPSIS Experiment

To facilitate the understanding and reproducibility of our findings, we will describe in this section the metrics used for comparing the fuzzy and crisp TOPSIS method and the experiment setup. We considered five scenarios that represent different user queries in the MEP system, each with a different need for the user performance criteria of energy consumption, accuracy and latency (i.e., the delay between the input of an event in the system and the event being processed and notified to the end-user). Each scenario is evaluated in all possible combinations of service worker alternatives with three, five and ten worker profiles, using 12 service worker profiles acquired from measurements of state-of-the-art DNN Object Detection models in a Cloud and Edge deployment environment. The TOPSIS methods are compared by considering the difference between the fuzzy TOPSIS ranking and the result of the crisp TOPSIS of all possible combinations of the crisp interpretation of the linguistic variable for the user performance requirements as the criteria weights. This process is repeated for all combinations of scenario and deployment environment, with 204820 evaluations executed and analysed for this paper.

1) Scenarios: We considered for all five scenarios the case of a user querying the MEP system for a person detection problem. That is, if a person is detected in a window of one frame of a video publisher during the query processing using a service worker DNN model for Object Detection, then this detection is immediately notified to the user in the form of a Video Event Knowledge Graph (VEKG) [21]. We considered the following scenario descriptions for the user performance criteria requirements, where the energy consumption and latency criteria, respectively, represent sustainability and responsiveness:

- **Scenario 1:** Highly sustainable, in which latency has a medium-high importance, and the accuracy importance is medium.
- **Scenario 2:** Medium-low sustainability, in which latency has a high importance, and the accuracy importance is medium-high.
- **Scenario 3:** Highly sustainable, but the other criteria have only low importance.
- **Scenario 4:** Highly accurate, but the other criteria have only low importance.
- **Scenario 5:** Highly responsive, but the criteria other than latency have only low importance.

The translation of these linguistic variables into their respective fuzzy representation is done through the linguistic terms scale for the criteria weights (see Table I). Therefore, the query using the fuzzy linguistic terms for Scenario 1 in the MEP system is defined as:

```
REGISTER QUERY AnyPersonFuzzyScenario1
OUTPUT K_GRAPH_JSON
CONTENT ObjectDetection
MATCH (p:person)
FROM SomeVideoInput
WITHIN TUMBLING_COUNT_WINDOW(1)
WITH_QOS
    energy_consumption = 'high_importance',
    accuracy = 'medium_importance',
    latency = 'medium_high_importance'
RETURN p
```

Meanwhile, for the same scenario (Scenario 1), the crisp queries would be similar to the fuzzy, with the only difference being the representation of the criteria weights, in which the user would need to interpret and decide on a crisp 10-point importance weight scale instead of using the linguistic terms. Therefore, for each scenario, we have up to 27 possible crisp interpretations of values based on the linguistic terms of each performance criterion weight. E.g., one possible interpretation of crisp weight values for Scenario 1 would be the values 9, 5 and 7 for the *high*, *medium* and *medium-high* importance of energy, accuracy and latency, respectively. Below is an example of how the requirements definitions would look like for this combination of crisp interpretation of weights for Scenario 1:

```
REGISTER QUERY AnyPersonCrisp1Scenario1
... // Same as before
WITH_QOS
    energy_consumption = 9,
    accuracy = 5,
    latency = 7
RETURN p
```

2) Deployment Environments: To understand how the number of alternatives can impact the ranking of the fuzzy and crisp TOPSIS methods, we executed the experiments in a setup with three, five and ten alternative service worker profiles. These profiles were previously created from the measurement of the energy consumption, accuracy and speed (throughput) of the DNN-based Object Detection models: SSD-MobilenetV1 (SSD) [22], Faster RCNN-InceptionV2 (Faster RCNN) [23], Faster RCNN-Inception-ResnetV2-Atrous (Faster RCNN-Atrous) [23]. All these models were pre-trained on the COCO 2017 image dataset [24]. These state-of-the-art models had their performance measured in a Cloud and Edge deployment environment. For the Cloud, a Dedicated Server with an Intel i9 8 Core CPU, 64 GB RAM, GeForce RTX 2080 TI 11GB GPU and 500 GB SSD was used, while for the Edge, the Jetson TX2 device was selected. Both of them were tested with and without GPU processing capabilities enabled [25]. These measurements led to a total of 12 unique service worker profiles (3 models $\times$ 2 Environments $\times$ 2 GPU/CPU), as seen in Table II.

Next, we applied the fuzzy pre-scaling to the worker profiles to get the alternatives' fuzzy representation according to the linguistic scale for the alternative ratings from Table I. This process was necessary for the fuzzy TOPSIS method and the pre-scaled crisp TOPSIS, except that the latter used a crisp 10-point scale. These 12 worker profiles for the service of Object Detection were used to provide multiple combinations of alternatives with three, five and ten workers to present the system with real-world data specific to the context of sustainable MEP on the Cloud and Edge.

A. Evaluation Metrics

We used a similar methodology as in previous works when comparing these MCDM methods concerning their contradiction rate (C-Rate) in their ranking results [12], [17], [26]. Moreover, to understand the limits of their ranking similarity, we analysed the C-Rate of the Top-1, Top-Half, and the complete rank (Top-N). Finally, we also compared the application of the pre-scaling in the crisp TOPSIS solution, as was done in [12], to evaluate the impact that this fuzzy-specific TOPSIS pre-scaling would have on the contradiction rates. We also considered the similarity between alternative ranking from [12]. However, since the similarity C-Rates were mainly identical to the exact C-Rates, we decided not to include them in this study.

The general formulas used to calculate the C-Rates are described below, in which k indicates up to what index of the rank with N elements the contradiction function $\text{Cont}(r^f, r^c, k)$ should be checked; r^f represents the ranking of the fuzzy TOPSIS method in a given scenario and deployment environment combination; C_f is the set of all crisp TOPSIS rankings (r^c) with an equivalent crisp interpretation of criteria weights to that used in r^f, using the same scenario and deployment environment.

$$\text{Cont}(r^f, r^c, k) = \begin{cases} 0 & \text{if } \forall 1 \geq i \leq k.\, r_i^f = r_i^c \ , \\ 1 & \text{otherwise.} \end{cases} \tag{3}$$

$$\text{C-Rate}(r^f, k) = \frac{\sum_{c \in C_f} \text{Cont}(r^f, r^c, k)}{\mid C_f \mid} \tag{4}$$

1) Top-1 C-Rate: This measures the rate of contradiction of only the best-ranked element in the resulting rank of the fuzzy and its crisp counterparts. This is defined as: $\text{C-Rate}(r^f, 1)$

2) Top-Half C-Rate: This identifies the contradiction rate of the top half (rounded up) of the best-ranked elements in ranking the fuzzy and crisp solutions. It is defined as: $\text{C-Rate}(r^f, \lceil N/2 \rceil)$

3) Top-N C-Rate: This represents the contradiction rate of the complete ranking of the TOPSIS methods. That is: there is a contradiction if any element in the ranked list of the fuzzy TOPSIS is in a different position in the crisp TOPSIS. This rate is defined by: $\text{C-Rate}(r^f, N)$

VI. Results and Discussion

Firstly, we will analyse the contradiction rates when comparing the fuzzy and crisp TOPSIS methods. Next, we will compare the results to the C-Rates when pre-scaling the alternative ratings in the crisp TOPSIS.

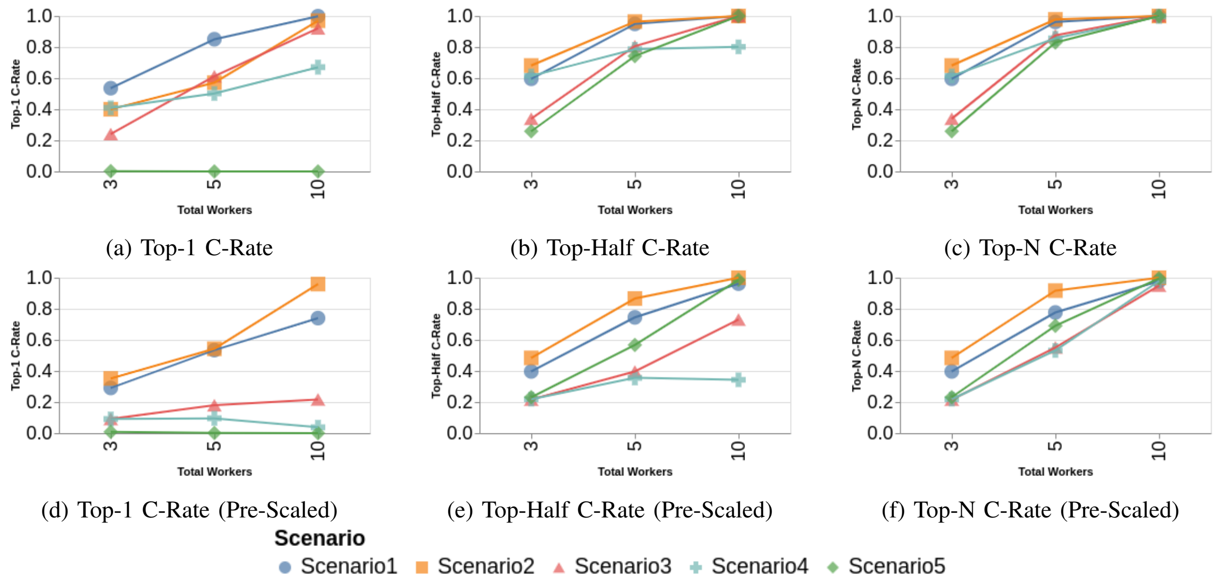
1) C-Rates for The Fuzzy and Crisp TOPSIS: By analysing the Top-1 C-Rate (see Figure 2a), we can observe that the C-Rates are significant in most scenarios except for Scenario 5. As expected, based on results from previous studies [12], [17], [26], these rates increase with the number of alternatives, with the lowest values being 24% in Scenario 3, with three alternatives, and 50% and 67%, in Scenario 4, with five and ten alternatives, respectively. Interestingly, in Scenario 5, the setup with three alternatives is the only one with a Top-1 C-Rate different from zero, with 0.22%. This odd behaviour could be attributed to the largely distinct throughput values of the available worker profiles, making it more likely that both methods would agree on the best worker in terms of throughput. That is: when mainly focusing on the latency criterion, there is only one set of worker profiles with close enough throughput values that might be affected by the uncertainty in the measurement of this metric in the specific case of three alternatives: the Faster RCNN-Atrous model deployed on the Cloud-CPU, Edge-GPU, and Edge-CPU, with 0.18, 0.14, and 0.02 FPS, respectively (see Table II). The increased Top-Half C-Rates further imply this possibility (see Figure 2b), with the lowest rates now being 25% and 74% in Scenario 5 with three and five alternatives, respectively, and 80% in Scenario 4 with ten alternatives.

Additionally, the high similarity between the Top-Half and Top-N C-Rates (see Figure 2c) shows that there are mostly no contradictions occurring exclusively on the lower half of the ranked elements, with the only exception being on Scenario 4 with ten alternatives, which has 19% of contradictions happening exclusively in this lower-half. In the context of sustainable MEP applications, this high Top-Half C-Rate could lead to a major discrepancy in the total energy consumption of the system since, in these applications, congestion controls are commonly applied to balance the load out of the top-ranked worker into the next best-ranked worker, once the first is overloaded with tasks. Moreover, the Top-Half C-Rate of 100% with ten alternatives, in all but one case, indicates that replacing the fuzzy with the crisp method would lead to a significantly high impact on sustainability for this case of MEP applications with high user demand.

TABLE II: Service Worker Profiles

Environment	Object Detection Model								
	SSD			Faster RCNN			Faster RCNN-Atrous		
	Energy* (Watts)	Accuracy* (mAP)	Throughput* (FPS)	Energy* (Watts)	Accuracy* (mAP)	Throughput* (FPS)	Energy* (Watts)	Accuracy* (mAP)	Throughput* (FPS)
Edge-GPU	6.6 (0.4)	21	1.33 (0.1)	8.6 (1.1)	28	0.67 (0.1)	14.3 (0.6)	37	0.14 (0.0)
Edge-CPU	8.3 (0.5)	21	4.43 (0.2)	12.0 (0.1)	28	0.42 (0.0)	12.0 (0.4)	37	0.02 (0.0)
Cloud-GPU	163.8 (19.9)	21	34.33 (1.1)	249.9 (26.8)	28	14.23 (3.2)	303.8 (17.5)	37	2.28 (0.43)
Cloud-CPU	188.0 (6.7)	21	45.6 (2.5)	230.6 (6.9)	28	2.89 (0.1)	222.0 (3.1)	37	0.18 (0.0)

*Average values with the standard deviation between parenthesis. mAP: Mean Average Precision. FPS: Frames Per Second



(a) Top-1 C-Rate (b) Top-Half C-Rate (c) Top-N C-Rate

(d) Top-1 C-Rate (Pre-Scaled) (e) Top-Half C-Rate (Pre-Scaled) (f) Top-N C-Rate (Pre-Scaled)

Fig. 2: First row (2a to 2c) shows C-Rates when comparing the fuzzy and crisp TOPSIS methods, while the second row (2d to 2f) has the C-Rates when using the pre-scaled crisp TOPSIS.

2) C-Rates Using Pre-scaled Crisp TOPSIS: We can observe from the second row in Figure 2 that applying a simple pre-scale step in the crisp method to make the input data of the crisp and fuzzy methods more similar leads to a significant reduction in the C-Rates compared with the previous scenarios. In the Top-1 C-Rate with ten alternatives (see Figure 2d), Scenarios 3 and 4 show a reduction of 71% and 64%, respectively, when compared to the C-Ratings without using the pre-scaling in the crisp method. Even in a setup with ten alternatives, scenarios 4 and 5 have very low Top-1 C-Rates of 3% and 0%, respectively. However, there are still high contradiction rates for Scenarios 1 and 2, which are the two scenarios that do not focus on only one criterion.

As mentioned before, in the context of sustainable MEP applications, it is necessary to consider more than just the top-1 element. Thus, we can observe an overall reduction in the Top-Half C-Rate, especially in Scenario 4 with ten alternatives, which shows a reduction of 46% (see Figure 2e). Additionally, the Top-N C-Rates with ten alternatives being mostly the same as before (see Figure 2f) indicates that it is more frequent the contradiction rates exclusively on the lower half of the rank when using the pre-scaled crisp TOPSIS. Nevertheless, we can still note that the Top-Half C-Rates are significantly higher, with the lowest values being 21% (Scenario 3 and 4), 35% (Scenario 4), and 34% (Scenario 4) for three, five and ten alternative worker profiles, respectively.

VII. Conclusion

This paper explored the similarity of the ranking results in the fuzzy and crisp TOPSIS methods in the context of uncertainty-aware user criteria requirements on the service selection in sustainable Multimedia Event Processing applications deployed on the Cloud and Edge. Our results show that the contradiction rate on the best-ranked element goes from 24% to 99% in most cases, with an even higher rate when comparing Top-Half and the complete rank list. Additionally, although the contradictions are less frequent compared with a pre-scaled crisp TOPSIS, the contradiction rates are still high, with at least 21% when comparing the Top-Half of the ranked elements. Our main conclusions are: Firstly, contrary to previous study [12], we note that there are many real-world scenarios in which the fuzzy TOPSIS cannot be replaced by its crisp counterpart without incurring a significant impact on the ranking results, which may affect the system sustainability efforts, especially when considering the ambiguities and uncertainties present in real-world variables and employing both benefit and cost criteria; secondly, the addition of a 10-point scale conversion to the alternatives in the crisp TOPSIS as to bring the input data closer to that of the fuzzy TOPSIS, although helpful in reducing the contradictions in 71% in one case, is not enough to replace the fuzzy method without some trade-offs in the expected ranking results, with contradiction rates mostly higher than 20%, and with a few scenarios going as high as 100%. In future works, we plan to evaluate and measure the impact these high contradiction rates have on the sustainability goals of an MEP system.

References

[1] M. Dayarathna, Y. Wen, and R. Fan, "Data center energy consumption modeling: A survey," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 732–794, 2015.

[2] A. Shehabi, S. Smith, D. Sartor, R. Brown, M. Herrlin, J. Koomey, E. Masanet, N. Horner, I. Azevedo, and W. Lintner, "United states data center energy usage report," 2016.

[3] L. Belkhir and A. Elmeligi, "Assessing ict global emissions footprint: Trends to 2040 & recommendations," *Journal of Cleaner Production*, vol. 177, pp. 448–463, 2018.

[4] J. Seo, S. Han, S. Lee, and H. Kim, "Computer vision techniques for construction safety and health monitoring," *Advanced Engineering Informatics*, vol. 29, no. 2, 2015.

[5] A. Aslam and E. Curry, "A survey on object detection for the internet of multimedia things (iomt) using deep learning and event-based middleware: Approaches, challenges, and future directions," *Image and Vision Computing*, vol. 106, p. 104095, 2021.

[6] E. Strubell, A. Ganesh, and A. McCallum, "Energy and policy considerations for deep learning in nlp," *arXiv preprint arXiv:1906.02243*, 2019.

[7] F. Gand, I. Fronza, N. El Ioini, H. R. Barzegar, S. Azimi, and C. Pahl, "A fuzzy controller for self-adaptive lightweight edge container orchestration," in *Proceedings of the 10th International Conference on Cloud Computing and Services Science-CLOSER*. SciTePress, 2020, pp. 79–90.

[8] A. Karanika, P. Oikonomou, K. Kolomvatsos, and T. Loukopoulos, "A demand-driven, proactive tasks management model at the edge," in *2020 IEEE international conference on fuzzy systems (FUZZ-IEEE)*. IEEE, 2020, pp. 1–8.

[9] S. G. Galán, M. Seddiki, R. J. P. de Prado, E. M. Expósito, A. Marchewka, and N. R. Reyes, "Relevance of using interpretability indexes for the design of schedulers in cloud computing systems," in *2020 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. IEEE, 2020, pp. 1–8.

[10] C.-T. Chen, "Extensions of the topsis for group decision-making under fuzzy environment," *Fuzzy sets and systems*, vol. 114, no. 1, pp. 1–9, 2000.

[11] C.-L. Hwang, K. Yoon, C.-L. Hwang, and K. Yoon, "Methods for multiple attribute decision making," *Multiple attribute decision making: methods and applications a state-of-the-art survey*, pp. 58–191, 1981.

[12] D. A. Pelta, M. T. Lamata, J. L. Verdegay, C. Cruz, and A. Salas, "Against artificial complexification: Crisp vs. fuzzy information in the topsis method," in *19th World Congress of the International Fuzzy Systems Association (IFSA), 12th Conference of the European Society for Fuzzy Logic and Technology (EUSFLAT), and 11th International Summer School on Aggregation Operators (AGOP)*. Atlantis Press, 2021, pp. 345–351.

[13] P. Jamshidi, C. Pahl, and N. C. Mendonça, "Managing uncertainty in autonomic cloud elasticity controllers," *IEEE Cloud Computing*, vol. 3, no. 3, pp. 50–60, 2016.

[14] S. Maheswari and G. Karpagam, "Enhancing fuzzy topsis for web service selection," *International Journal of Computer Applications in Technology*, vol. 51, no. 4, pp. 344–351, 2015.

[15] M. Masdari and H. Khezri, "Service selection using fuzzy multi-criteria decision making: a comprehensive review," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 2, pp. 2803–2834, 2021.

[16] E. Triantaphyllou and C.-T. Lin, "Development and evaluation of five fuzzy multiattribute decision-making methods," *international Journal of Approximate reasoning*, vol. 14, no. 4, pp. 281–310, 1996.

[17] B. Ceballos, M. T. Lamata, D. A. Pelta, and R. R. Yager, "Crisp vs. fuzzy data in multicriteria decision making: the case of the vikor method," in *Advances in Fuzzy Logic and Technology 2017: Proceedings of: EUSFLAT-2017–The 10th Conference of the European Society for Fuzzy Logic and Technology, September 11–15, 2017, Warsaw, Poland IWIFSGN'2017–The Sixteenth International Workshop on Intuitionistic Fuzzy Sets and Generalized Nets, September 13–15, 2017, Warsaw, Poland, Volume 1 10*. Springer, 2018, pp. 455–465.

[18] L. A. Zadeh, "Fuzzy logic, neural networks, and soft computing," in *Fuzzy sets, fuzzy logic, and fuzzy systems: selected papers by Lotfi A Zadeh*. World Scientific, 1996, pp. 775–782.

[19] T. J. Ross, *Fuzzy logic with engineering applications*. John Wiley & Sons, 2005.

[20] J. B. Cabral, N. A. Luczywo, and J. L. Zanazzi, "Scikit-criteria: Colección de métodos de análisis multi-criterio integrado al stack científico de Python," in *XLV Jornadas Argentinas de Informática e Investigación Operativa (45JAIIO)- XIV Simposio Argentino de Investigación Operativa (SIO) (Buenos Aires, 2016)*, 2016, pp. 59–66. [Online]. Available: http://45jaiio.sadio.org.ar/sites/default/files/Sio-23.pdf

[21] E. Curry, D. Salwala, P. Dhingra, F. A. Pontes, and P. Yadav, "Multimodal event processing: A neural-symbolic paradigm for the internet of multimedia things," *IEEE Internet of Things Journal*, 2022.

[22] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[23] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[24] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[25] F. A. Pontes and E. Curry, "Cloud-edge microservice architecture for dnn-based distributed multimedia event processing," in *Advances in Service-Oriented and Cloud Computing: International Workshops of ESOCC 2020, Heraklion, Crete, Greece, September 28–30, 2020, Revised Selected Papers 8*. Springer, 2021, pp. 65–72.

[26] E. Triantaphyllou and S. H. Mann, "An examination of the effectiveness of multi-dimensional decision-making methods: A decision-making paradox," *Decision support systems*, vol. 5, no. 3, pp. 303–312, 1989.