mathematics

Article

RMVC: A Validated Algorithmic Framework for Decision-Making Under Uncertainty

Abdurrahman Dayioglu *, Fatma Ozen Erdogan and Basri Celik

Department of Mathematics, Bursa Uludag University, Nilufer, Bursa 16059, Turkiye; fatmaozen@uludag.edu.tr (F.O.E.); basri@uludag.edu.tr (B.C.)
* Correspondence: dayioglu@uludag.edu.tr

Abstract

The reliability of decision-making algorithms within soft set theory is fundamentally constrained by their underlying membership functions. Traditional binary approaches overlook the implicit connections between the attributes a candidate possesses and those it lacks—connections that can be inferred from the wider candidate pool. To address this core challenge, this paper puts forward the Relational Membership Value Calculation (RMVC), an algorithmic framework whose core is a fine-grained relational membership function. Our approach moves beyond binary logic to capture these nuanced interrelationships. We provide a rigorous theoretical analysis of the proposed algorithm, including its computational complexity and robustness, which is validated through a comprehensive sensitivity analysis. Crucially, a comparative analysis using the Gini Index quantitatively demonstrates that our method provides significantly higher granularity and discriminatory power on a representative case study. The RMVC is implemented as an open-source Python program, providing a foundational tool to enhance the reasoning capabilities of AI-driven decision support and expert systems.

Keywords: soft set theory; relational membership function; decision-making algorithms; uncertainty modeling; decision support systems; open-source AI tools

MSC: 03-04; 03E72; 03E75; 68T37; 65K05; 91B06

check for updates

Academic Editor: Hsien-Chung Wu

Received: 31 July 2025
Revised: 16 August 2025
Accepted: 18 August 2025
Published: 21 August 2025

Citation: Dayioglu, A.; Erdogan, F.O.; Celik, B. RMVC: A Validated Algorithmic Framework for Decision-Making Under Uncertainty. *Mathematics* **2025**, *13*, 2693. https://doi.org/10.3390/math13162693

1. Introduction

Soft set theory provides a robust and flexible framework for addressing problems characterized by uncertainty, ambiguity, or incomplete information. Unlike conventional set theories, which often struggle with imprecise data, soft set theory offers a more nuanced approach by associating elements of a universal set with a parameter set. This association creates a collection of sets that effectively captures the variability and imprecision inherent in real-world problems. First proposed by Molodtsov in 1999 [1] as a generalization and alternative to fuzzy set theory and fuzzy logic, soft set theory has since been extensively refined and expanded. Its capacity to model and manage complex decision-making scenarios where information is partially known or inherently vague has established it as a powerful tool across a wide range of disciplines, including artificial intelligence, economics, engineering, and data science.

Molodtsov developed soft set theory to fill the gaps left by classical and fuzzy set theories, which often struggled to handle complex uncertainty and ambiguity in practical situations. Soft set theory offered a pioneering approach to problems involving imprecise or incomplete information by introducing a more flexible and generalized framework. Following its introduction, the theory garnered widespread attention, leading to an acceleration of mathematical research and the exploration of its applications across numerous disciplines. In computer science, soft sets proved helpful for handling data uncertainty, while in decision theory, they provided enhanced models for complex, multi-criteria decision-making processes. Similarly, fields like data mining and artificial intelligence benefited from their capacity to process ambiguous information more effectively than traditional methods. Economists and biologists also found value in the theory, using it to model uncertainty in economic systems and biological data. Consequently, soft set theory has evolved into a versatile tool for addressing the intricate challenges of uncertain environments.

In the 2000s, soft set theory saw significant growth and refinement as researchers expanded its foundational concepts and explored new applications. Early on, Maji et al. [2] showcased practical applications for soft set theory, and the following year [3], they introduced essential concepts such as equality and subset relations within this framework. Chen et al. [4] worked on simplifying soft sets by reducing the number of parameters needed, aiming to streamline the computational aspects of soft sets. Ali et al. [5] added new operations between soft sets, further expanding the theory's scope. Kong et al. [6] introduced the concept of normal parameter reduction, along with algorithms that made this approach more applicable in decision-making processes using soft sets. In 2010, Cagman and Enginoglu [7] reexamined key operations between soft sets, working on the product of soft sets and introducing the uni-int decision-making method, which enhanced decision processes under uncertain conditions. The theory continued to evolve in the following years: Peng and Yang, in [8], developed algorithms based on interval-valued fuzzy soft sets for multi-criteria decision-making in stochastic environments. By 2019, their work extended into the use of inverse fuzzy soft sets in decision-making [9], further solidifying soft set theory as a valuable tool for tackling complex, real-world problems.

Soft sets provide an effective mathematical model for decision-making processes focused on selecting the best option due to their ability to express objects that provide parameters within a soft set. This structure has been successfully utilized in numerous studies aimed at eliminating uncertainties. Recent works [10–25] have shown the importance of soft sets as a tool in managing uncertainty and decision-making processes.

Since 2020, studies on soft set theory have made significant progress, particularly in uncertainty management and multi-criteria decision-making (MCDM) problems. During this period, soft set theory has been expanded, and hybrid models have been developed. Structures like fuzzy soft sets and interval-valued fuzzy soft sets have been effective in optimizing decision-making processes.

Additionally, research on filling missing data and parameter reduction has allowed soft sets to be used more efficiently with large datasets. These studies have developed various algorithms for predicting missing data, enabling decision-making processes to proceed despite this missing information.

These developments demonstrate that soft set theory is a flexible and powerful tool in decision sciences and uncertainty management.

While foundational methods, such as the algorithm proposed by Maji and Roy [3], provide a viable approach to decision-making with soft sets, they can suffer from a loss of sensitivity due to the summation process, potentially masking nuanced differences between alternatives. The inherent limitations of binary membership values in classical soft set decision-making have been noted in the literature. Some approaches, for instance, address this issue by redefining the membership concept to be relational [14]. Our work also addresses this fundamental challenge but introduces a distinct mathematical formulation through the Θ function. This function not only captures relational aspects but is embedded within a framework that allows for its quantitative validation. Unlike conceptual models, we empirically demonstrate the effectiveness of our approach in enhancing decision granularity using the Gini Index and provide a comprehensive computational and sensitivity analysis, offering a robust and testable solution to the problem. We develop and present the Relational Membership Value Calculation (RMVC), a novel algorithm that offers a more granular and sensitive scoring mechanism. The primary contribution of this work lies not only in the enhanced theoretical framework but also in its practical implementation as a fully automated, open-source computational tool, thereby increasing the transparency, reproducibility, and applicability of soft set theory in real-world decision-making scenarios. The enhanced granularity and robustness of the RMVC algorithm make it a promising component for integration into larger intelligent systems. Specifically, its ability to provide nuanced rankings could significantly improve the reliability of AI-driven applications such as multi-criteria expert systems, medical diagnostic tools, and personalized recommender systems, where ambiguity in decision-making can lead to suboptimal outcomes. The remainder of this paper is organized as follows:

Section 2 reviews the fundamental concepts of soft set theory.

Section 3 introduces the RMVC algorithm in detail, presents its core relational membership function, Θ, and provides a rigorous analysis of its theoretical properties, including computational complexity and robustness to data sparsity.

Section 4 is dedicated to the validation of our algorithmic framework, where we first conduct a sensitivity analysis to test the model's stability under perturbation.

Following this, Section 5 grounds our theoretical claims by applying the RMVC framework to an illustrative decision-making scenario. Within the context of this real-world problem, we use the Gini Index to provide quantitative evidence that our algorithm achieves higher granularity compared to established methods.

Finally, Section 6 concludes this paper with a summary of our findings and suggests directions for future research.

2. Preliminaries

This section presents the basic definitions and results of fuzzy sets, soft sets, and inverse soft sets required in the following sections. Detailed explanations of soft sets and inverse soft sets can be found in [5,7,26]. To improve clarity and ensure consistency in mathematical notation, we have made some adjustments to the definitions and representations found in the literature.

Definition 1. *Let $\mathcal{U}$ be a universal set. A fuzzy set F on $\mathcal{U}$ is characterized by a membership function $\mu : \mathcal{U} \to [0,1]$, where for each $x \in \mathcal{U}$, the value $\mu(x)$ indicates the degree to which x belongs to F. This fuzzy set is denoted as follows:*

$$F = \{(x, \mu(x)) : x \in \mathcal{U}\}.$$

Here, μ is called the *membership function* of $\mathcal{U}$, and the value $\mu(x)$ is called the *grade of membership* of $x \in \mathcal{U}$.

Definition 2. *Assume that $\mathcal{U}$ is a set of elements and E is a set of parameters. An ordered pair (A, Φ) is called a soft set over $\mathcal{U}$, where $A \subseteq E$ and Φ is a mapping given by*

$$\Phi : A \longrightarrow P(\mathcal{U})$$

where $P(\mathcal{U})$ is the power set of $\mathcal{U}$.

In this study, we specifically focus on the case where $A = E$, and the decision process involves only a single decision-maker.

A soft set (E, Φ) over $\mathcal{U}$ can be represented as the set of ordered pairs

$$(E, \Phi) = \{(e, \Phi(e)) : e \in E, \Phi(e) \in P(\mathcal{U})\}.$$

Definition 3. *Assume that $\mathcal{U}$ is a set of elements and E is a set of parameters. An ordered pair $(\mathcal{U},\psi)$ is called an inverse soft set (ISS) over E, where ψ is a mapping given by*

$$\psi : \mathcal{U} \longrightarrow P(E).$$

An inverse soft set $(\mathcal{U}, \psi)$ over E can be represented by the set of ordered pairs

$$(\mathcal{U}, \psi) = \{(u, \psi(u)) : u \in \mathcal{U}, \psi(u) \in P(E)\}.$$

As established by Cetkin [26], each soft set can be uniquely represented as an inverse soft set, and furthermore, each fuzzy set generates an inverse soft set.

3. A Different Approach to Decision-Making Under Uncertainty via Python Code

In this section, we redefine the relational and inverse relational membership functions to enhance sensitivity and accuracy in decision-making. Furthermore, to obtain more exact and reliable results in soft set theory, we developed a program called the "Relational Membership Value Calculator" (RMVC) using Python 3.13.3.

Definition 4. *Let $\mathcal{U} = \{1, 2, 3, \dots, n\}$ be a universal set and let $E = \{e_1, e_2, e_3, \dots, e_m\}$ be a set of parameters. Let (E, Φ) be a soft set over $\mathcal{U}$. For each $1 \le i \le m$, and for any $k \in \mathcal{U} \setminus \Phi(e_i)$, the relational membership value of k with respect to $\Phi(e_i)$ is defined via the mapping*

$$\Theta_{e_i} : \mathcal{U} \setminus \Phi(e_i) \longrightarrow [0, 1].$$

This mapping, called the relational membership function, assigns a membership value to each element k not belonging to $\Phi(e_i)$ and given by

$$\Theta_{e_i}(k) = \frac{1}{|\Phi(e_i)| \cdot (m-1)} \sum_{e_t \in E \setminus \{e_i\}} \sum_{j \in \Phi(e_i)} \delta_{e_t}(k, j),$$

where $1 \le j, k \le n$, $1 \le i, t \le m$, and $n, m \ge 2$.

Here, for $e_t \in E \setminus \{e_i\}$, the Kronecker delta function $\delta_{e_t} : [\mathcal{U} \setminus \Phi(e_i)] \times \Phi(e_i) \longrightarrow \{0, 1\}$ is defined as

$$\delta_{e_t}(k, j) = \begin{cases} 1; & k \in \Phi(e_t) \wedge j \in \Phi(e_t), \\ 0; & \textit{otherwise}. \end{cases}$$

Thus, the relational membership function measures how strongly an element k (not originally in $\Phi(e_i)$) is connected to the elements of $\Phi(e_i)$ through other parameters.

Theorem 1. *The proposed relational membership function, $\Theta_{e_i}(k)$, is consistent and monotonic.*

Proof. The proof is structured in two parts, demonstrating the properties of consistency and monotonicity based on the function's definition.

1. Consistency: This property requires the function to produce values within a predictable, well-defined range. We will prove that the function $\Theta_{e_i}(k)$ is always bounded within the interval $[0, 1]$.

The function is defined as

$$\Theta_{e_i}(k) = \frac{1}{|\Phi(e_i)| \cdot (m-1)} \sum_{e_t \in E \setminus \{e_i\}} \sum_{j \in \Phi(e_i)} \delta_{e_t}(k, j)$$

Lower bound: The Kronecker delta function, $\delta_{e_t}(k, j)$, can only take values of 0 or 1. Therefore, the double summation in the numerator must be greater than or equal to zero. The denominator, which consists of the cardinality of a set ($|\Phi(e_i)|$) and the number of other parameters ($m - 1$), is always a positive integer (since $m \geq 2$). Thus, the entire expression is non-negative:

$$\Theta_{e_i}(k) \geq 0$$

Upper bound: To find the maximum possible value, consider the numerator. The outer sum runs over $m - 1$ parameters in $E \setminus \{e_i\}$, and the inner sum runs over $|\Phi(e_i)|$ elements. In the most extreme case, $\delta_{e_t}(k, j)$ would be 1 for every single term in the sum. The maximum value of the double summation is therefore the total number of terms, which is $|\Phi(e_i)| \cdot (m - 1)$. Substituting this maximum value into the function gives

$$\Theta_{e_i}(k)_{max} = \frac{|\Phi(e_i)| \cdot (m-1)}{|\Phi(e_i)| \cdot (m-1)} = 1$$

Since the numerator can never exceed this value, we have

$$\Theta_{e_i}(k) \leq 1$$

Combining these bounds confirms that $0 \leq \Theta_{e_i}(k) \leq 1$, proving that the function is consistent.

2. Monotonicity: In the context of this function, monotonicity relates specifically to the co-occurrence relationship. The property implies that as the relational connection of an element k to the set $\Phi(e_i)$ increases, its membership value $\Theta_{e_i}(k)$ must also increase. The relational connection is measured by the sum of δ values.

Let the total co-occurrence count be $C(k) = \sum_{e_t \in E \setminus \{e_i\}} \sum_{j \in \Phi(e_i)} \delta_{e_t}(k, j)$. The function can be written as

$$\Theta_{e_i}(k) = \frac{C(k)}{|\Phi(e_i)| \cdot (m-1)}$$

The denominator is a positive constant for a given e_i. Therefore, $\Theta_{e_i}(k)$ is directly proportional to the co-occurrence count $C(k)$. If any event occurs that increases the relational connection (e.g., for a previously non-matching pair (k, j) under a parameter e_t, a change causes them to co-occur, so $\delta_{e_t}(k, j)$ changes from 0 to 1), the value of $C(k)$ increases. Consequently, the value of $\Theta_{e_i}(k)$ also increases.

This direct proportionality demonstrates that the function is monotonic with respect to the strength of the relational connections, which is the core concept of the proposed function. □

Example 1. *Let $\mathcal{U} = \{1, 2, 3, 4, 5\}$ be a set of elements and $E = \{e_1, e_2, e_3, e_4\}$ be a set of parameters. If $\Phi(e_1) = \{1, 2, 3, 5\}, \Phi(e_2) = \{2, 4, 5\}, \Phi(e_3) = \{1, 3, 4\}, \Phi(e_4) = \{1, 2, 5\}$, then the corresponding soft set (E, Φ) is given by*

$$(E, \Phi) = \{(e_1, \{1, 2, 3, 5\}), (e_2, \{2, 4, 5\}), (e_3, \{1, 3, 4\}), (e_4, \{1, 2, 5\})\}.$$

We now calculate all the relational membership values for this soft set.

First we consider the parameter e_1 *(for* $i = 1$*). The only element in* $\mathcal{U}$ *but not in* $\Phi(e_1)$ *is* $k = 4$*. So the value* $\Theta_{e_1}(4)$ *is calculated as follows:*

$$\begin{aligned}
\Theta_{e_1}(4) &= \frac{1}{4.3}[\delta_{e_2}(4,j) + \delta_{e_3}(4,j) + \delta_{e_4}(4,j)], \qquad \forall j \in \Phi(e_1) \\
&= \frac{[0+1+0+1] + [1+0+1+0] + [0+0+0+0]}{12} \\
&= \frac{1}{3}
\end{aligned}$$

The remaining relational values were calculated in a similar manner. The complete set of results is as follows:

$$\begin{gathered}
\Theta_{e_2}(1) = \frac{5}{9},\ \Theta_{e_2}(3) = \frac{1}{3}, \\
\Theta_{e_3}(2) = \frac{4}{9},\ \Theta_{e_3}(5) = \frac{4}{9}, \\
\Theta_{e_4}(3) = \frac{4}{9},\ \Theta_{e_4}(4) = \frac{1}{3}.
\end{gathered}$$

These results can be compiled into a matrix for a more comprehensive representation as shown in Table 1.

Table 1. Membership matrix format.

	1	2	3	4	5
e_1	1	1	1	$\frac{1}{3}$	1
e_2	$\frac{5}{9}$	1	$\frac{1}{3}$	1	1
e_3	1	$\frac{4}{9}$	1	1	$\frac{4}{9}$
e_4	1	1	$\frac{4}{9}$	$\frac{1}{3}$	1

The discriminative power of the RMVC algorithm is inherently linked to the relational richness of the input data. The core of the method, the relational membership value Θ, quantifies the strength of an alternative's relationship with other alternatives for each parameter set. This calculation relies on the co-occurrence of alternatives as measured by the δ function.

In highly sparse data environments—where parameter sets ($\Phi(e_i)$) contain few members and have minimal overlap—the opportunities for such co-occurrences diminish significantly. Consequently, the numerator of the Θ formula, representing the sum of relational interactions, tends toward smaller values. For any non-member of a given parameter set, its calculated relational score will therefore converge towards zero.

This leads to a critical boundary condition. As data becomes sparser, the resulting Θ matrix becomes increasingly polarized and dominated by binary-like values: "1" for members and values approaching "0" for non-members. In such a state, the nuanced, relational membership values that give the RMVC method its depth are reduced, and the algorithm's behavior begins to approximate a simple inclusion–exclusion check rather than a rich relational analysis. While the algorithm remains structurally sound, its ability to differentiate finely between non-members is consequently reduced. This highlights an important dependency: the full potential of the RMVC method is realized when the input soft set is sufficiently dense to generate a strong relational signal.

To facilitate the practical application of our method (algorithmic framework), we provide a reference implementation in Python. The following code snippet illustrates the core of the RMVC algorithm, especially demonstrating how user input for the universal set $\mathcal{U}$ is processed. The complete open-source program is publicly available for researchers and practitioners around the globe who are interested in further details or direct application. The repository can be accessed at https://github.com/DrDayioglu/RMVC.git (20 August 2025).

The initial step involves defining the universal set $\mathcal{U}$. The following Python function is used to capture this input from the user:

```python
def get_input_set(prompt):
"""Helper function to get a set from user input."""
raw_input = input(prompt)
elements = raw_input.split()
return set(elements)

U = get_input_set("Enter the elements of set U (separate with
                                spaces): ")
```

Next the parameter set E and the corresponding subsets of $\mathcal{U}$ must be defined, since the soft set mapping, ϕ, assigns a subset of $\mathcal{U}$ (an element of the power set, $P(\mathcal{U})$) to each parameter in E; the program must first be capable of generating the power set of $\mathcal{U}$ to construct these relationships.

```python
def power_set(s):
"""Generate the power set of a given set s."""
return list(chain.from_iterable(combinations(s,r) for r in range(len(
                                  s)+1)))
P_power_set = power_set(U)
def get_subset(power_set):
"""Helper function to get a subset of the power set from user input.
                                  """
subset = []
n = 1
while True:
element = input(f"The elements of e_{n} : (Press enter to terminate):
                                  ")
if element == "":
break
subset.append(tuple(element.split()))
n += 1
return~subset

E_named = get_subset(P_power_set)
```

Next we define the sum part of the delta function, which is crucial for this newly presented method.

```
def delta_function(e_name, E_named):
# Take the considered set and the elements
e_set = E_named[e_name]  # All elements in the selected set
not_in_e_set = U - e_set  # Elements not in the selected set
results = {}

# For every element not in the selected set
for element in not_in_e_set:
total_sum = 0
# Create pairs with elements in the selected set
for other_element in e_set:
for other_e_set in E_named.values():
# Check if this pair is present in any other set
if {element, other_element}.issubset(other_e_set):
total_sum += 1
break  # Exit loop after incrementing the counter
results[element] = total_sum
# Save the result for each element

return results
```

To ensure data consistency and improve readability, the input data is first converted into a list of sets using the following code snippet:

```
  E = [set(subset) for subset in E_named]
```

Next, each parameter set is mapped to a unique identifier $e_1, e_2, \ldots, e_m$ to facilitate programming access. This is achieved with the following code.

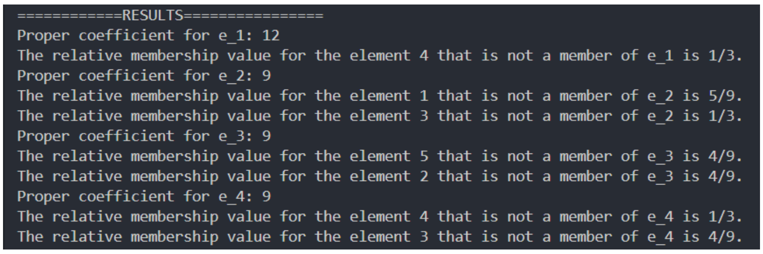
```
    e_name = {f"e_{i+1}": subset for i, subset in enumerate(E)}
    E_keys = e_name.keys()
```

The core calculation is then executed. The following loop iterates through each parameter set to complete the final relational membership values.

```
    for e_key in E_keys:
    delta_results = delta_function(e_key, e_name)
    g_coeff = len(e_name[e_key]) * (len(E_named) - 1)
    print(f"Proper coefficient for {e_key}:", g_coeff)

    for u, delta_sum in delta_results.items():
    M = Fraction(delta_sum, g_coeff)
    print(f"The relative membership value for the element {u} that is
                                        not a member of {e_key} is {M}.")
```

The complete output generated by the *"Relational Membership Value Calculator"* (*RMVC*) for this example is given in Figure 1.



```
============RESULTS===============
Proper coefficient for e_1: 12
The relative membership value for the element 4 that is not a member of e_1 is 1/3.
Proper coefficient for e_2: 9
The relative membership value for the element 1 that is not a member of e_2 is 5/9.
The relative membership value for the element 3 that is not a member of e_2 is 1/3.
Proper coefficient for e_3: 9
The relative membership value for the element 5 that is not a member of e_3 is 4/9.
The relative membership value for the element 2 that is not a member of e_3 is 4/9.
Proper coefficient for e_4: 9
The relative membership value for the element 4 that is not a member of e_4 is 1/3.
The relative membership value for the element 3 that is not a member of e_4 is 4/9.
```

Figure 1. Results of RMVC for Example 1.

3.1. Computational Complexity Analysis

To evaluate the scalability of the RMVC algorithm for large-scale soft sets, we analyze its theoretical time complexity. Let $n = |\mathcal{U}|$ be the number of elements (candidates), $m = |E|$ be the number of parameters (criteria), and k_{avg} be the average cardinality of a parameter set $\Phi(e_i)$.

The algorithm's complexity is determined by three primary computational steps:

1. **Calculation of the membership matrix ($\boldsymbol{\mathcal{M}_{m \times n}}$):** This is the most computationally intensive step. To calculate a single $\Theta_{e_i}(j)$ value, the algorithm iterates through all other $m-1$ parameters and, for each, iterates through the elements of $\Phi(e_i)$. This results in a complexity of approximately $O(m \cdot k_{avg})$ for a single entry. Since there are $n \times m$ entries to compute, the total complexity for this step is $O(n \cdot m \cdot (m \cdot k_{avg})) = O(n \cdot m^2 \cdot k_{avg})$.
2. **Calculation of final scores:** This step involves summing the n columns of the $m \times n$ matrix, which has a complexity of $O(n \cdot m)$.
3. **Selection of the optimal candidate:** Finding the maximum value among n scores takes $O(n)$ time.

The overall time complexity is determined by the most demanding step, which is the calculation of the membership matrix. Therefore, the computational complexity of the RMVC algorithm is approximately $\mathbf{O}(\mathbf{n} \cdot \mathbf{m}^2 \cdot \mathbf{k}_{\mathbf{avg}})$.

This analysis indicates that the algorithm's performance scales linearly with the number of candidates (n) but quadratically with the number of parameters (m). This suggests that the RMVC is well-suited for problems with a very large number of candidates but may become computationally expensive for applications involving an extremely high number of evaluation criteria.

3.2. Sensitivity Analysis

A crucial attribute of any clustering algorithm is its robustness against small perturbations in the input data. For the RMVC, the input consists of the parameter sets $\Phi(e_i)$. Sensitivity analysis for our model, therefore, involves examining how the final clustering output reacts to minor changes within these sets, such as the addition or removal of a single element.

3.2.1. Relational Similarity Matrix

The relational membership function, $\Theta_{e_i}(u)$, produces a **relational membership value** that indicates the association of an element u with a specific parameter set $\Phi(e_i)$. To obtain a complete relational profile of an element, we collect these values across all parameter sets. We define this profile as the **Relational Membership Vector**.

For any element $u \in \mathcal{U}$, its membership vector $\boldsymbol{\Theta}_{\mathbf{u}}$ is an $|E|$-dimensional vector where each component is the element's relational membership value with respect to one parameter set:

$$\boldsymbol{\Theta}_{\mathbf{u}} = [\Theta_{e_1}(u), \Theta_{e_2}(u), \ldots, \Theta_{e_{|E|}}(u)]$$

Clustering requires us to measure the pairwise similarity between any two elements, u and v. A natural way to achieve this is by comparing their respective membership vectors, $\mathbf{\Theta_u}$ and $\mathbf{\Theta_v}$. We define a similarity matrix S, where the entry S_{uv} is computed by measuring the average L1-distance between these vectors. A smaller distance implies higher similarity. The formula is defined as

$$S_{uv} = 1 - \frac{1}{|E|} \sum_{i=1}^{|E|} |\Theta_{e_i}(u) - \Theta_{e_i}(v)|$$

Here, $S_{uv} = 1$ indicates that the elements have identical membership vectors (perfect similarity), while a value approaching 0 indicates high dissimilarity. This matrix S provides the foundational input for the final graph-partitioning step and for the sensitivity analysis that follows.

Let us consider a small perturbation in a single parameter set $\Phi(e_i)$, resulting in a new set $\Phi'(e_i)$. This change propagates through our model in a clear sequence:

1. **Change in membership values:** A perturbation in a single parameter set, say from $\Phi(e_i)$ to $\Phi'(e_i)$, propagates its effect to all relational membership values across the system, not just one. This happens in two ways:
   - **Direct effect (on $\mathbf{\Theta_{e_i}}$):** The membership function associated directly with the changed parameter, $\Theta_{e_i}(k)$, is the most impacted. Both its denominator, which contains the term $|\Phi(e_i)|$, and its numerator, which sums over $j \in \Phi(e_i)$, are altered. This requires a full recalculation of $\Theta_{e_i}(k)$ for all elements k.
   - **Indirect effect (on $\mathbf{\Theta_{e_t}}$ for all $\boldsymbol{t \neq i}$):** The relational membership value for any other parameter, $\Theta_{e_t}(k)$, is also affected. Its calculation involves a summation over all parameters $e_u \in E \setminus \{e_t\}$. Since $t \neq i$, this collection of parameters necessarily includes e_i. Therefore, the $\delta_{e_i}(k, j)$ term within the sum will now use the perturbed set $\Phi'(e_i)$. This change in a single delta term will cause a small but non-zero change in the value of $\Theta_{e_t}(k)$, rippling the effect of the initial perturbation throughout the entire system.
2. **Change in relational similarity matrix:** Since the relational similarity matrix S is constructed from all Θ values, the changes described above will lead to a modified similarity matrix S'.
3. **Potential change in final partition:** The final clustering partition P^* is derived from S. A change from S to S' may or may not be significant enough to alter the final partition P^*.

The relational nature of the Θ function provides a degree of inherent robustness. The membership value is an average of co-occurrence information calculated over all other parameters ($m - 1$) and all elements within a set. Therefore, a single change (one δ value flipping from 0 to 1, or the addition/removal of one element) has a diluted effect on the final Θ value rather than causing a catastrophic shift.

To quantitatively measure this sensitivity, one can perform the following analysis:

- Let S be the original similarity matrix computed with the initial parameter sets $\{\Phi(e_i)\}_{i=1}^{m}$.
- Introduce a single perturbation, for instance, by adding one element to a single set, to get $\{\Phi'(e_i)\}$.
- Compute the new similarity matrix S'.
- The magnitude of the change can be measured by the **Frobenius norm** of the difference between the matrices: Sensitivity $= ||S - S'||_F$. A lower sensitivity score indicates higher robustness.

- Furthermore, one can compare the final clustering partitions P^* (from S) and $P^{*'}$ (from S') using an external validation index like the **Adjusted Rand Index (ARI)**. An ARI score of 1 would signify that the small perturbation had no effect on the final clustering result, demonstrating perfect stability for that specific change.

This analytical framework allows for a systematic evaluation of the algorithm's stability, confirming that the RMVC is not overly sensitive to minor variations in its input parameterization, a desirable trait for real-world applications.

3.2.2. A Practical Case Study of Sensitivity

To provide a concrete illustration of the theoretical sensitivity framework, we apply it step-by-step to the setup from Example 1. This practical analysis demonstrates how to compute the similarity matrix, introduce a change to the system, and quantitatively measure the impact on the final clustering outcome.

Step 1: Establishing the Baseline with the Original Data

Our first task is to establish a baseline result using the original, unperturbed data from Example 1. This involves computing the relational similarity matrix, S, using the formula defined in the previous section.

To ensure complete transparency and reproducibility, this section provides a detailed, step-by-step calculation of each unique entry in the similarity matrix S for the setup in Example 1. This detailed demonstration is intended to showcase the mechanics of the methodology once. For the sake of abbreviation, in subsequent sections, similar intermediate calculation steps will be omitted and only the final results will be presented.

Step 1: Constructing Relational Membership Vectors.

First, we construct the relational membership vectors, $\boldsymbol{\Theta}_\mathbf{u}$, for every element $u \in \mathcal{U}$. All subsequent calculations are based on these vectors:

$\boldsymbol{\Theta}_\mathbf{1} = [1, 0.556, 1, 1]$
$\boldsymbol{\Theta}_\mathbf{2} = [1, 1, 0.444, 1]$
$\boldsymbol{\Theta}_\mathbf{3} = [1, 0.556, 1, 0.667]$
$\boldsymbol{\Theta}_\mathbf{4} = [0.333, 1, 1, 0.333]$
$\boldsymbol{\Theta}_\mathbf{5} = [1, 1, 0.444, 1]$

Step 2: Pairwise similarity calculations.

We now calculate the similarity S_{uv} for each pair of elements using the following formula:

$$S_{uv} = 1 - \frac{1}{4} \sum_{i=1}^{4} |\Theta_{e_i}(u) - \Theta_{e_i}(v)|$$

To illustrate the process, we give a detailed calculation for a few representative entries:

- **Calculating S_{12}:**

  The sum of absolute differences is $|1-1| + |0.556-1| + |1-0.444| + |1-1| = 0 + 0.444 + 0.556 + 0 = 1$.
  $S_{12} = 1 - (1/4) = 1 - 0.250 = \mathbf{0.750}$.

- **Calculating S_{24}:**

  The sum of absolute differences is $|1-0.333| + |1-1| + |0.444-1| + |1-0.333| = 0.667 + 0 + 0.556 + 0.667 = 1.890$.
  $S_{24} = 1 - (1.890/4) = 1 - 0.472 = \mathbf{0.528}$.

The remaining entries of the similarity matrix S were calculated using the same procedure. This process is repeated for all pairs. The completed results form the final similarity matrix, which is presented in the next step.

Step 3: Assembling the final similarity matrix.

Compiling all the calculated similarity values yields the final similarity matrix, $S = [S_{ij}], i = 1, 2, 3, 4, 5$:

S					
	1	2	3	4	5
1	1	0.750	0.917	0.555	0.750
2	0.750	1	0.667	0.528	1
3	0.917	0.667	1	0.639	0.667
4	0.555	0.528	0.639	1	0.528
5	0.750	1	0.667	0.528	1

Applying a graph-partitioning method to this matrix, we can derive the initial clustering partition, P^*. For this analysis, we define a "strong tie" as any connection with a similarity score greater than 0.85. An analysis of the strong ties in matrix S reveals the following:

- The similarity S_{13} is 0.917 (>0.85), indicating a strong tie. This forms the core cluster: $\{1, 3\}$.
- The similarity S_{25} is 1 (>0.85), also indicating a strong tie. This forms a second distinct cluster: $\{2, 5\}$.
- No other similarity values exceed the 0.85 threshold. Element "4" does not form a strong tie with any other element and is therefore considered an outlier.

This analysis yields the following initial partition P^*:

$$P^* = \{\{1, 3\}, \{2, 5\}, \{4\}\}.$$

This partition consists of two distinct, cohesive clusters and a single outlier element.

Step 2: Defining and Applying the Perturbation

Next, a minimal, localized perturbation is introduced to the system to test its stability. We modify the parameter set $\Phi(e_3)$ by removing the element "4". The new, perturbed set is denoted as $\Phi'(e_3) = \{1, 3\}$. All other parameter sets remain unchanged.

Step 3: Calculating the Post-Perturbation Outcome

Following this change, the relational membership vectors ($\boldsymbol{\Theta}_{\mathbf{u}}$) and, consequently, the entire similarity matrix must be recalculated. After performing all calculations with the perturbed data, we compute the new similarity matrix, S'.

S′					
	1	2	3	4	5
1	1	0.750	1	0.472	0.750
2	0.750	1	0.750	0.528	1
3	1	0.750	1	0.472	0.750
4	0.472	0.528	0.472	1	0.528
5	0.750	1	0.750	0.528	1

Step 4: Quantitative Analysis and Interpretation

The final step is to quantitatively compare the pre- and post-perturbation results to measure the algorithm's stability.

Frobenius norm: To measure the overall change in the similarity structure, we calculate the Frobenius norm of the difference between the original matrix S and the perturbed matrix S':

$$||S - S'||_F = \sqrt{\sum_{i,j} (S_{ij} - S'_{ij})^2} \approx 0.333$$

The low value indicates that the perturbation did not cause a drastic shift; the overall relational structure between elements remained highly stable.

Adjusted Rand Index (ARI): To measure the effect on the final partition, we compare the original partition, P^*, with the new partition, $P^{*'}$, derived from S'. Applying the same partitioning logic (a similarity threshold > 0.85) to S' yields an identical partition:

$$P^{*'} = \{\{1,3\},\{2,5\},\{4\}\}$$

The new partition is identical to the original. The ARI score, which measures the agreement between two data clusterings, confirms this perfect stability:

$$\text{ARI}(P^*, P^{*'}) = 1.0$$

An ARI score of 1.0 signifies that the final output was completely unaffected by the perturbation. This practical demonstration validates the robustness of the RMVC algorithm, showcasing its resilience against minor variations in the input data.

4. A New Decision-Making Approach

In this section, building on the new technical formulation for soft sets presented in the third section, we introduce a new algorithmic method for decision-making in uncertain environments, basically the outlines of our Python code for the *RMVC*. We develop the Algorithm 1 that applies our new formulation to this context.

Algorithm 1 Determine the best choice in a given SS.

Input: Let $\mathcal{U} = \{1, 2, 3, \ldots, n\}$ be the set of elements, let $E = \{e_1, e_2, e_3, \ldots, e_m\}$ be the set of parameters. Given a soft set (E, Φ), where $m, n \geq 2$.

Step 1: Computing all relational membership values using the Θ function as defined in the new formula.

Step 2: Constructing the matrix $\mathcal{M}_{m \times n}$ which contains the membership values derived from $\Omega_{\Theta_{e_i}(r)}$ where $r = 1, 2, \ldots, n$, $i = 1, 2, \ldots, m$:

$$\mathcal{M}_{m \times n} = \begin{bmatrix} \Omega_{\Theta_{e_1}(1)} & \Omega_{\Theta_{e_1}(2)} & \Omega_{\Theta_{e_1}(3)} & \cdots & \Omega_{\Theta_{e_1}(n)} \\ \Omega_{\Theta_{e_2}(1)} & \Omega_{\Theta_{e_2}(2)} & \Omega_{\Theta_{e_2}(3)} & \cdots & \Omega_{\Theta_{e_2}(n)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \Omega_{\Theta_{em}(1)} & \Omega_{\Theta_{em}(2)} & \Omega_{\Theta_{em}(3)} & \cdots & \Omega_{\Theta_{em}(n)} \end{bmatrix}$$

and $\Omega_{\Theta_{e_i}(r)}$ is defined as

$$\Omega_{\Theta_{e_i}(r)} = \begin{cases} 1 & ; \quad r \in \Phi(e_i) \\ \Theta_{e_i}(r) ; & \text{otherwise.} \end{cases}$$

Here, $\Omega : E \times \mathcal{U} \longrightarrow [0, 1]$ represents the membership function.

Step 3: Calculating the score, $S(r)$ of elements $r \in \mathcal{U}$ using the following formula:

$$S(r) = \sum_{j=1}^{m} \Omega_{\Theta_{e_j}(r)}.$$

Step 4: Selecting the element r with the highest score, namely, the one for which $S(r)$ is maximized.

A fundamental aspect of the proposed framework is the choice of using a raw sum for the final score aggregation ($S(r)$). This decision was made deliberately, prioritizing interpretability and parameter-free objectivity in this foundational presentation of the RMVC model while acknowledging that more complex aggregation methods exist.

4.1. Score Resolution and Tie-Breaking Mechanism

A critical attribute of any decision-making algorithm is its ability to produce a clear and decisive ranking and and to minimize the occurrence of ties. The proposed RMVC scoring mechanism holds a significant theoretical advantage in this regard compared to traditional soft set approaches.

Many conventional methods in soft set theory, such as the one proposed by Maji and Roy, rely on scoring systems based on integer counts. In such systems, a candidate's score is typically the total number of criteria they satisfy. This restricts the set of possible scores to a small, discrete set of integers, $\{0, 1, \ldots, m\}$. When the range of possible outcomes is limited, the probability of different candidates achieving the exact same score is inherently higher.

In contrast, the RMVC methodology is based on the relational membership function, Θ, which produces rational numbers as values. The final score for a candidate, $S(r)$, is the sum of these granular values. This process maps the candidates to a much larger and denser set of possible scores. Because the image set of the scoring function is significantly larger and denser, the probability of a coincidental tie between two distinct candidates is substantially reduced. While an absolute guarantee against ties is not possible, the structural nature of the RMVC renders them a rare occurrence.

To ensure the algorithm is robust and deterministic even in the rare case of a tie, the final selection is formally defined as a two-tiered process:

1. Primary criterion: The primary selection criterion is the relational score $S(r)$. The optimal candidate, u^*, is the one that maximizes the score:

$$u^* = \arg\max_{u_i \in \mathcal{U}} S(i)$$

2. Secondary (tie-breaking) criterion: If two or more candidates share an identical maximal score, a secondary tie-breaking criterion is applied. For this, we use a simpler choice value, k_i, representing the total count of parameter sets to which candidate u_i belongs. The candidate with the higher k_i value among the tied candidates is chosen. If the tie persists even after this secondary criterion, the candidates are considered equally optimal and can be presented as a set of co-optimal choices.

Example 2. *Let $\mathcal{U} = \{1, 2, 3, 4, 5, 6, 7\}$ be a set of candidates and $E = \{e_1, e_2, e_3, e_4, e_5, e_6\}$ be a set of evaluation criteria for a specific job position. For each $i = 1, 2, \ldots, 6$, the element e_i in E represents a required attribute, such as "higher education", "experience", "foreign language proficiency", etc.*

The decision-making problem is to select the optimal candidate for the position.

The soft set (E, Φ) is defined by the following:

$$\Phi(e_1) = \{1, 2, 3, 6\}, \Phi(e_2) = \{2, 4, 6\}, \Phi(e_3) = \{1, 3, 7\},$$
$$\Phi(e_4) = \{3, 5, 6\}, \Phi(e_5) = \{1, 5, 7\}, \Phi(e_6) = \{2, 3, 6\}.$$

This can be expressed as the set of ordered pairs:

$$\begin{aligned}(E,\Phi) = \{&(e_1,\{1,2,3,6\}),(e_2,\{2,4,6\}),(e_3,\{1,3,7\}),\\ &(e_4,\{3,5,6\}),(e_5,\{1,5,7\}),(e_6,\{2,3,6\})\}\end{aligned}$$

We proceed by calculating all relational membership values for (E,Φ), which will then be compared with the output of the RMVC software implementation.

Since candidates $\{4,5,7\}$ do not satisfy the first criterion e_1, their corresponding relational membership values are calculated as

$$\Theta_{e_1}(4) = 1/10, \Theta_{e_1}(5) = 3/20, \Theta_{e_1}(7) = 3/20.$$

Similarly, since the candidates $\{1,3,5,7\}$ do not satisfy the second criterion e_2, their corresponding relational membership values are given by

$$\Theta_{e_2}(1) = 2/15, \Theta_{e_2}(3) = 1/3, \Theta_{e_2}(5) = 1/15, \Theta_{e_2}(7) = 0.$$

The complete set of remaining relational memberships values is as follows:

$$\Theta_{e_3}(2) = \frac{1}{5}, \Theta_{e_3}(4) = 0, \Theta_{e_3}(5) = \frac{1}{5}, \Theta_{e_3}(6) = \frac{4}{15},$$

$$\Theta_{e_4}(1) = \frac{4}{15}, \Theta_{e_4}(2) = \frac{1}{3}, \Theta_{e_4}(4) = \frac{1}{15}, \Theta_{e_4}(7) = \frac{2}{15}$$

$$\Theta_{e_5}(2) = \frac{1}{15}, \Theta_{e_5}(3) = \frac{4}{15}, \Theta_{e_5}(4) = 0, \Theta_{e_5}(6) = \frac{2}{15}$$

$$\Theta_{e_6}(1) = \frac{4}{15}, \Theta_{e_6}(4) = \frac{2}{15}, \Theta_{e_6}(5) = \frac{2}{15}, \Theta_{e_6}(7) = \frac{1}{15}$$

This completes Step 1.

Following Step 2, we prepare the matrix as

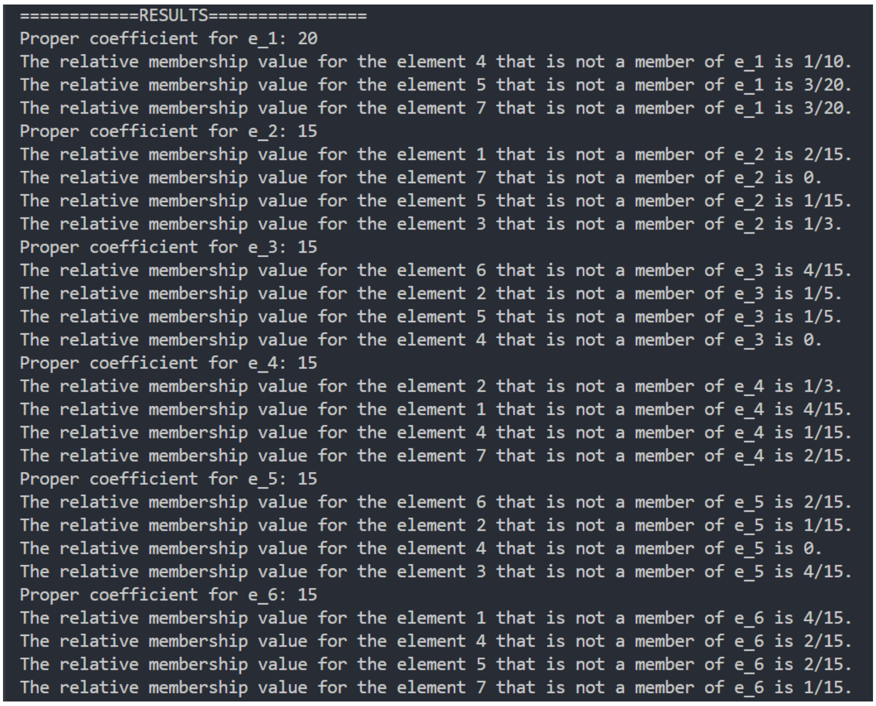
$$\begin{bmatrix} 1 & 1 & 1 & 0.10 & 0.15 & 1 & 0.15 \\ 0.13 & 1 & 0.33 & 1 & 0.07 & 1 & 0 \\ 1 & 0.2 & 1 & 0 & 0.2 & 0.27 & 1 \\ 0.27 & 0.33 & 1 & 0.07 & 1 & 1 & 0.13 \\ 1 & 0.07 & 0.27 & 0 & 1 & 0.13 & 1 \\ 0.27 & 1 & 1 & 0.13 & 0.13 & 1 & 0.07 \end{bmatrix}$$

In Step 3, the final scores are calculated for each candidate:

$$S(1) = 3.67, S(2) = 3.6, S(3) = 4.6, S(4) = 1.30, S(5) = 2.55, S(6) = 4.4, S(7) = 2.35$$

In Step 4, the optimal choice is determined by selecting the candidate with the maximum score. In this case, $S(3)$ is the maximum value, indicating that candidate 3 is the optimal choice.

These results were cross-validated using our software implementation. Feeding the same input data into the RMVC tool yielded identical scores, as shown in Figure 2.



```
============RESULTS===============
Proper coefficient for e_1: 20
The relative membership value for the element 4 that is not a member of e_1 is 1/10.
The relative membership value for the element 5 that is not a member of e_1 is 3/20.
The relative membership value for the element 7 that is not a member of e_1 is 3/20.
Proper coefficient for e_2: 15
The relative membership value for the element 1 that is not a member of e_2 is 2/15.
The relative membership value for the element 7 that is not a member of e_2 is 0.
The relative membership value for the element 5 that is not a member of e_2 is 1/15.
The relative membership value for the element 3 that is not a member of e_2 is 1/3.
Proper coefficient for e_3: 15
The relative membership value for the element 6 that is not a member of e_3 is 4/15.
The relative membership value for the element 2 that is not a member of e_3 is 1/5.
The relative membership value for the element 5 that is not a member of e_3 is 1/5.
The relative membership value for the element 4 that is not a member of e_3 is 0.
Proper coefficient for e_4: 15
The relative membership value for the element 2 that is not a member of e_4 is 1/3.
The relative membership value for the element 1 that is not a member of e_4 is 4/15.
The relative membership value for the element 4 that is not a member of e_4 is 1/15.
The relative membership value for the element 7 that is not a member of e_4 is 2/15.
Proper coefficient for e_5: 15
The relative membership value for the element 6 that is not a member of e_5 is 2/15.
The relative membership value for the element 2 that is not a member of e_5 is 1/15.
The relative membership value for the element 4 that is not a member of e_5 is 0.
The relative membership value for the element 3 that is not a member of e_5 is 4/15.
Proper coefficient for e_6: 15
The relative membership value for the element 1 that is not a member of e_6 is 4/15.
The relative membership value for the element 4 that is not a member of e_6 is 2/15.
The relative membership value for the element 5 that is not a member of e_6 is 2/15.
The relative membership value for the element 7 that is not a member of e_6 is 1/15.
```

Figure 2. Results of RMVC for Example 2.

Although the general sensitivity analysis framework discussed in Section 3 already validated the robustness of the RMVC algorithm, we will now perform a similar perturbation on the decision-making problem of Example 3 to reiterate and reinforce this principle. The objective is to validate whether the optimal choice, candidate 3, maintains its rank under a minor perturbation. To demonstrate a case of stability, we introduce a specific perturbation and analyze its consequences.

As a test case, we assume that candidate 5 now satisfies the criterion e_1 ("higher education"), modifying the parameter set to $\Phi'(e_1) = \{1,2,3,5,6\}$. This change requires a full recalculation of relational scores. Table 2 compares the original scores, all with the new scores generated by the RMVC software following this perturbation.

Table 2. Score Comparison.

Candidate	Original Score ($S(i)$)	New Score ($S'(i)$)	Score Change
1	3.67	3.73	↑
2	3.60	3.73	↑
3	**4.60** (Rank 1)	**4.67** (Rank 1)	↑
4	1.30	1.28	↓
5	2.55	3.87	↑
6	4.40	4.47	↑
7	2.35	2.36	↑

As the results demonstrate, the perturbation had a significant and appropriate impact on the score of the directly targeted candidate (candidate 5), increasing it from 2.55 to 3.87. Furthermore, due to the model's relational nature, the scores of most other candidates were also slightly adjusted.

Despite these adjustments, the ranking of the top candidates remained stable. Candidate 3 retained the highest score, and thus the optimal decision was unchanged. This analysis confirms that while the model is appropriately responsive to minor changes, selecting the optimal candidate exhibits strong robustness against this type of data variation.

4.1.1. ComparisonAnalysis

To contextualize the performance of our method, we compare its results with those obtained from the traditional algorithm proposed by Maji and Roy [3], using the decision-making problem from Example 2. The algorithm from [3] produces the following candidate ranking, which includes several ties:

$$\mathbf{3} = \mathbf{6} > \mathbf{1} = \mathbf{2} > \mathbf{5} = \mathbf{7} > \mathbf{4}.$$

In contrast, our RMVC algorithm yields a fully resolved ranking with no ties:

$$\mathbf{3} > \mathbf{6} > \mathbf{1} > \mathbf{2} > \mathbf{5} > \mathbf{7} > \mathbf{4}.$$

This immediate result demonstrates that the RMVC method provides a more granular and decisive ordering of candidates.

In this study, we define *granularity* as the ability of a scoring method to discriminate between alternatives by reducing ties and producing a more finely distributed score set. To further deepen the analysis of score granularity, we now compare the structural characteristics of the score distributions generated by the RMVC method and the traditional Maji–Roy approach. For this purpose, we treat the set of scores from each method as a discrete probability distribution by normalizing them (i.e., dividing each score by the total sum). We then compute the Gini Index for each resulting distribution to provide a standardized measure of concentration. Unlike the conventional interpretation where higher Gini values are associated with greater inequality, our framework interprets a **lower Gini Index** as an indicator of higher granularity, since it reflects a less concentrated and more evenly spread distribution of scores.

This analysis is conducted on the decision-making problem of Example 2, as it provides a set of scores for each candidate.

4.1.2. Analysis of the Maji–Roy (Choice Value) Distribution

The traditional method produces scores based on the count of criteria satisfied by each candidate:

1. **Raw scores (k_i):** The set of scores for the seven candidates is $\{3, 3, 4, 1, 2, 4, 2\}$.
2. **Normalization:** The sum of these scores is $3 + 3 + 4 + 1 + 2 + 4 + 2 = 19$. We normalize each score by dividing by 19 to get a probability distribution:

$$P_{\text{Maji-Roy}} = \{0.158, 0.158, 0.211, 0.053, 0.105, 0.211, 0.105\}$$

3. **Gini Index calculation:** We first sort the normalized probabilities:

$$\{0.053, 0.105, 0.105, 0.158, 0.158, 0.211, 0.211\}.$$

Applying the Gini Index formula to this distribution yields

$$G_{\text{Maji-Roy}} \approx \mathbf{0.210}$$

4.1.3. Analysis of the RMVC Distribution

Next, we perform the same analysis on the scores generated by our proposed RMVC method:

1. **Raw scores ($S(i)$):** As presented in Example 2, the scores are
$$\{3.67, 3.60, 4.60, 1.30, 2.55, 4.40, 2.35\}.$$
2. **Normalization:** The sum of these scores is 22.47. We normalize to get the probability distribution:
$$P_{\text{RMVC}} = \{0.163, 0.160, 0.205, 0.058, 0.113, 0.196, 0.105\}$$
3. **Gini Index calculation:** We first sort the normalized probabilities:
$$\{0.058, 0.105, 0.113, 0.160, 0.163, 0.196, 0.205\}.$$
Applying the Gini Index formula to this distribution yields
$$G_{\text{RMVC}} \approx \mathbf{0.192}$$

4.1.4. Conclusion of Comparative Analysis

The comparative results are summarized below in Table 3.

Table 3. Gini Index comparison.

Score Distribution	Gini Index
Traditional (Maji–Roy)	0.210
Proposed (RMVC)	0.192

The results provide consistent quantitative evidence supporting our central claim. The Gini Index for the RMVC scores (0.192) is lower than that of the traditional count-based method (0.210). Even in this small-scale example, this represents a relative reduction in score concentration of approximately 9%. This suggests that the relational membership function maps candidates to a more distributed score space, structurally reducing the probability of ties and enabling a more discriminative basis for decision-making. While the effect size is modest in absolute terms, its presence even in a minimal test case indicates the method's potential for enhanced granularity in broader applications.

Example 3. *To demonstrate the flexibility of the RMVC method, we now analyze the inverse problem from Example 2. In this inverse soft set (ISS) formulation, the roles of the sets are reversed. $E = \{e_1, e_2, e_3, e_4, e_5, e_6\}$ now represents the set of candidates, and $\mathcal{U} = \{1, 2, 3, 4, 5, 6, 7\}$ represents the set of criteria (e.g., "higher education", "experience", "foreign language", etc.). The decision-making problem is now to determine which candidate is most strongly supported by the available pool of criteria.*

If $\psi(1) = \{e_1, e_3, e_5\}, \psi(2) = \{e_1, e_2, e_6\}, \psi(3) = \{e_1, e_3, e_4, e_6\}, \psi(4) = \{e_2\}, \psi(5) = \{e_4, e_5\}, \psi(6) = \{e_1, e_2, e_4, e_6\}, \psi(7) = \{e_3, e_5\}$, then the ISS$(\psi, \mathcal{U})$ is written by

$$\begin{aligned}(\psi, \mathcal{U}) = \{&(1, \{e_1, e_3, e_5\}), (2, \{e_1, e_2, e_6\}), (3, \{e_1, e_3, e_4, e_6\}),\\ &(4, \{e_2\}), (5, \{e_4, e_5\}), (6, \{e_1, e_2, e_4, e_6\}), (7, \{e_3, e_5\})\}\end{aligned}$$

Utilizing the RMVC allows us to derive the complete set of relational membership values. When we organize these values into a matrix format as outlined in our algorithm, we subsequently obtain the matrix presented below.

$$\begin{bmatrix} 1 & 0.11 & 1 & 0.22 & 1 & 0.22 \\ 1 & 1 & 0.17 & 0.28 & 0.06 & 1 \\ 1 & 0.21 & 1 & 1 & 0.17 & 1 \\ 0.33 & 1 & 0 & 0.17 & 0 & 0.33 \\ 0.25 & 0.08 & 0.25 & 1 & 1 & 0.17 \\ 1 & 1 & 0.17 & 1 & 0.08 & 1 \\ 0.25 & 0 & 1 & 0.17 & 1 & 0.08 \end{bmatrix}$$

As expected from the model's asymmetric nature, the resulting unified membership matrix is not the transpose of the matrix from the forward problem. The final scores for each candidate (now parameters) are calculated as

$$S(e_1) = 4.83, S(e_2) = 3.4, S(e_3) = 3.58, S(e_4) = 3.83, S(e_5) = 3.31, S(e_6) = 3.81.$$

The maximum score is $S(e_1)$, indicating that candidate e_1 is the optimal choice in the inverse formulation. This result suggests that from the perspective of the available criteria, candidate e_1 has the strongest overall profile.

4.2. A Small Note on the Asymmetry of the Relational Model

In classical soft set theory, which uses a binary incidence matrix (where entries are 1 or 0), the matrix of the inverse soft set is indeed the transpose of the forward soft set's matrix. This is because the underlying relationship is a simple, symmetric "belongs to" check.

However, the RMVC method does not operate on this simple incidence matrix. Instead, it generates a relational membership matrix, $\mathcal{M}_{m\times n}$, where each entry $\Theta_{e_i}(u_j)$ is a calculated, real-valued score. The calculation of this score is fundamentally asymmetric. The formula computes the membership of a candidate u_j with respect to a parameter e_i by analyzing the co-occurrence patterns of u_j across the entire universe of other parameters ($E \setminus \{e_i\}$).

In an "inverse" relational problem—calculating the membership of a parameter e_i with respect to a candidate u_j—the roles of $\mathcal{U}$ and E would be interchanged in the formula. The calculation would then be based on the co-occurrence patterns of e_i across the universe of other candidates ($\mathcal{U} \setminus \{u_j\}$). This would lead to a completely different set of calculations and a resulting matrix that is not a simple transpose of the former relational matrix.

This asymmetry does not impact model consistency; on the contrary, it is the source of the model's novelty and power. It ensures that the "view" from the perspective of parameters (how candidates relate to a criterion) and the "view" from the perspective of candidates (how criteria relate to a candidate) are distinct and rich with contextual information. This is a deliberate and core feature of the relational model, not an inconsistency.

We now compare the ranking produced by our inverse RMVC analysis with the ranking from a traditional count-based approach [3] for the same inverse problem. The count-based method, which relies on integer scores, results in a ranking with extensive ties:

$$e_1 > e_2 = e_3 = e_4 = e_5 = e_6.$$

In sharp contrast, the RMVC algorithm produces a fully resolved and granular ranking:

$$e_1 > e_4 > e_6 > e_3 > e_2 > e_5.$$

This comparison again highlights the discriminative power of the RMVC method. By moving beyond simple integer counts to a relational, real-valued scoring system, the algorithm effectively breaks ties and provides a more detailed and useful ordering. This enhanced granularity is a direct result of the model's design and is critical for nuanced decision-making in complex scenarios.

5. Application to a Real-World Benchmark Case

To validate the practical efficiency of the proposed RMVC method, we apply it to a well-established benchmark problem from the MCDM literature. This approach not only demonstrates the algorithm's applicability to a real-world scenario but also allows for a direct comparison of its performance against established methods.

5.1. Benchmark Dataset: The Automotive Selection Problem

The dataset used is the classic automotive selection problem presented by Yoon [27] in his seminal paper "A Reconciliation among Discrete Compromise Solutions." He sourced it from Fleischer's book [28]. This problem has since become a standard benchmark, used by numerous researchers to test and compare new decision-making methods. The problem involves ranking eight car models based on 16 different criteria.

The original decision matrix, as presented in Yoon [27] (p. 284, Table 2), is shown in Table 4, below. This table includes the raw performance values for each car across all criteria, as well as the criterion weights assigned in the original study.

Table 4. The original decision matrix from Yoon (1987) [27].

Criterion	Type	Weight	A1	A2	A3	A4	A5	A6	A7	A8
			Ford	Chevrolet	Plymouth	Mercury	Ford	Cadillac	Volvo	Peugeot
			Pinto	Nova	Duster	Monarch	Granada	Seville	244 DL	504
1. List price (USD)	Cost	0.216	2769	3222	3526	4565	9948	3679	3269	9214
2. Operating cost (USD/yr)	Cost	0.038	1170	1540	1790	1980	2000	1540	1540	2000
3. Resale value after 5 yr	Benefit	0.004	1000	1200	1200	2000	4000	1200	1200	4000
4. Maintainability	Benefit	0.027	5.0	5.0	7.0	2.5	6.0	7.0	5.0	6.0
5. City mileage (mpg)	Benefit	0.162	18.0	12.0	10.0	9.0	11.0	19.0	12.0	10.0
6. Highway mileage (mpg)	Benefit	0.162	26.0	19.0	16.0	18.0	16.0	28.0	19.0	15.0
7. Acceleration (0-60; s)	Cost	0.032	14.7	19.1	10.8	13.0	13.7	16.2	15.3	13.5
8. Braking (60-0; ft)	Cost	0.002	180.0	165.0	155.6	141.6	178.0	170.0	165.0	190.6
9. Ride	Benefit	0.092	4.3	8.6	8.6	5.7	10.0	5.7	8.6	5.7
10. Handling	Benefit	0.017	7.0	7.0	7.0	7.0	4.0	4.0	7.0	1.0
11. Maneuverability	Benefit	0.022	7.0	7.0	8.0	6.0	5.0	6.0	7.0	3.0
12. Trunk space (cu. ft)	Benefit	0.009	6.3	13.0	14.3	20.4	13.6	10.9	16.0	19.3
13. Front-seat room	Benefit	0.108	7.5	7.5	10.0	7.5	10.0	7.5	7.5	10.0
14. Front-seat comfort	Benefit	0.108	7.1	8.6	7.5	7.1	7.1	8.6	7.1	7.1
15. Rear-seat room	Benefit	0.001	7.5	7.5	10.0	7.5	10.0	10.0	7.5	10.0
16. Rear-seat comfort	Benefit	0.001	5.0	7.5	7.5	7.5	7.5	10.0	7.5	7.5

Data from G. A. Fleischer [28].

5.2. Data Preprocessing: From Raw Data to a Comprehensive Soft Set

The original dataset is extensive, comprising 16 distinct criteria. To provide a robust and challenging test for our RMVC method, we select a comprehensive subset of 12 criteria. This expanded set moves beyond a minimal selection and allows for a more nuanced analysis, testing our method against a rich and diverse set of attributes.

The goal of this selection is to create a balanced model covering the primary aspects of vehicle evaluation: long- and short-term economics, performance and efficiency, safety, comfort, and practicality. As our RMVC method does not use the criterion weights, demonstrating its effectiveness on a larger, more complex set of unweighted criteria is a key validation step.

The selected criteria, now grouped by their evaluation category, are as follows:

- **Economic factors (three criteria):** We include the primary "List price" (C1) and the long-term "Operating cost" (C2) to cover different cost aspects. To complete the economic picture, we also include "Resale value" (C3), a low-weight benefit criterion.
- **Performance and efficiency (four criteria):** We select both "City mileage" (C5) and "Highway mileage" (C6) for a full efficiency profile. For performance, we include the opposing metrics of "Acceleration" (C7) and "Braking" (C8), both of which are cost-type criteria requiring transformation.
- **Driving dynamics and comfort (three criteria):** We include "Ride" (C9) for comfort and "Handling" (C10) for vehicle dynamics. To represent interior comfort, we add "Front-seat room" (C13).
- **Practicality and maintenance (two criteria):** We include "Maintainability" (C4) and "Trunk space" (C12) as key indicators of a vehicle's day-to-day usability.

This curated subset of 12 parameters provides a rigorous test case for the RMVC method. We now denote these parameters as $e_1, \ldots, e_{12}$ for consistency with our soft set notation. The raw data for this comprehensive subset is shown below in Table 5.

Table 5. Selected raw data for 12 criteria (Yoon, 1987) [27].

Parameter	Type	A1	A2	A3	A4	A5	A6	A7	A8
e_1: Price (USD)	Cost	2769	3222	3526	4565	9948	3679	3269	9214
e_2: Operating cost (USD/yr)	Cost	1170	1540	1790	1980	2000	1540	1540	2000
e_3: Resale value (USD)	Benefit	1000	1200	1200	2000	4000	1200	1200	4000
e_4: Maintainability	Benefit	5.0	5.0	7.0	2.5	6.0	7.0	5.0	6.0
e_5: City mileage (mpg)	Benefit	18.0	12.0	10.0	9.0	11.0	19.0	12.0	10.0
e_6: Highway mileage (mpg)	Benefit	26.0	19.0	16.0	18.0	16.0	28.0	19.0	15.0
e_7: Acceleration (s)	Cost	14.7	19.1	10.8	13.0	13.7	16.2	15.3	13.5
e_8: Braking (ft)	Cost	180.0	165.0	155.6	141.6	178.0	170.0	165.0	190.6
e_9: Ride	Benefit	4.3	8.6	8.6	5.7	10.0	5.7	8.6	5.7
e_{10}: Handling	Benefit	7.0	7.0	7.0	7.0	4.0	4.0	7.0	1.0
e_{11}: Front-seat room	Benefit	7.5	7.5	10.0	7.5	10.0	7.5	7.5	10.0
e_{12}: Trunk space (cu. ft)	Benefit	6.3	13.0	14.3	20.4	13.6	10.9	16.0	19.3

To apply our RMVC method, all parameters must be of the benefit type (i.e., higher values are better). The four cost-type parameters (e_1, e_2, e_7, e_8) are converted into benefit values using a scaled reciprocal transformation ($1/x$).

To improve the readability of the transformed data and ensure the values are of a comparable magnitude to the other criteria, we use scaling factors of 1000 or 100 in the transformation (i.e., $1000/x$ or $100/x$). This linear scaling does not change the internal ranking of the alternatives within the criterion but makes the resulting table more intuitive. The benefit-type parameters remain unchanged. The resulting "benefit table," which serves as the basis for our analysis, is presented in Table 6.

Table 6. Transformed benefit table for 12 criteria (with scaled values).

Parameter	A1	A2	A3	A4	A5	A6	A7	A8
e_1: Price (1000/USD)	0.361	0.310	0.284	0.219	0.101	0.272	0.306	0.109
e_2: Operating cost (1000/USD)	0.855	0.649	0.559	0.505	0.500	0.649	0.649	0.500
e_3: Resale value (USD)	1000	1200	1200	2000	4000	1200	1200	4000
e_4: Maintainability	5.0	5.0	7.0	2.5	6.0	7.0	5.0	6.0
e_5: City mileage (mpg)	18.0	12.0	10.0	9.0	11.0	19.0	12.0	10.0
e_6: Highway mileage (mpg)	26.0	19.0	16.0	18.0	16.0	28.0	19.0	15.0
e_7: Acceleration (100/s)	6.80	5.24	9.26	7.69	7.30	6.17	6.54	7.41
e_8: Braking (100/ft)	0.556	0.606	0.643	0.706	0.562	0.588	0.606	0.525
e_9: Ride	4.3	8.6	8.6	5.7	10.0	5.7	8.6	5.7
e_{10}: Handling	7.0	7.0	7.0	7.0	4.0	4.0	7.0	1.0
e_{11}: Front-seat room	7.5	7.5	10.0	7.5	10.0	7.5	7.5	10.0
e_{12}: Trunk space (cu. ft)	6.3	13.0	14.3	20.4	13.6	10.9	16.0	19.3

5.3. Constructing the Soft Set

With all 12 parameters converted to a uniform benefit format as shown in Table 6, we can now define the soft set parameter sets, $\Phi(e_i)$. To ensure maximum objectivity and methodological consistency, we employ a single, data-driven thresholding rule across parameters.

The "Above Average" rule: For each of the 12 parameters, we calculate the arithmetic mean of the benefit values for all eight alternatives. An alternative is considered to "possess" the parameter (i.e., belong to the set $\Phi(e_i)$) if its benefit value is strictly greater than the mean for that parameter. This unified approach eliminates subjective threshold-setting and grounds the entire preprocessing phase in the internal structure of the data.

For example, for the first parameter e_1 ("Economical Price"), the arithmetic mean of the benefit values in Table 6 is calculated as 0.245. The alternatives with a score strictly greater than this mean are $\{A_1, A_2, A_3, A_6, A_7\}$. Therefore, the resulting parameter set is

$$\Phi(e_1) = \{A_1, A_2, A_3, A_6, A_7\}$$

Applying this same data-driven rule to the remaining 11 parameters yields the complete soft set. The results are summarized in Table 7 below.

Table 7. Parameter ID and resulting soft set.

Parameter ID	Parameter Name	Calculated Mean	Resulting Soft Set $\Phi(e_i)$
e_1	Price	0.245	$\{A_1, A_2, A_3, A_6, A_7\}$
e_2	Operating cost	0.608	$\{A_1, A_2, A_6, A_7\}$
e_3	Resale value	1975	$\{A_4, A_5, A_8\}$
e_4	Maintainability	5.438	$\{A_3, A_5, A_6, A_8\}$
e_5	City mileage	12.625	$\{A_1, A_6\}$
e_6	Highway mileage	19.625	$\{A_1, A_6\}$
e_7	Acceleration	7.051	$\{A_3, A_4, A_5, A_8\}$
e_8	Braking	0.599	$\{A_2, A_3, A_4, A_7\}$
e_9	Ride	7.15	$\{A_2, A_3, A_5, A_7\}$
e_{10}	Handling	5.5	$\{A_1, A_2, A_3, A_4, A_7\}$
e_{11}	Front-seat room	8.438	$\{A_3, A_5, A_8\}$
e_{12}	Trunk space	14.225	$\{A_3, A_4, A_7, A_8\}$

This transparent and objective preprocessing yields a well-defined soft set, which is represented in a binary matrix format in Table 8. This matrix now serves as the final input for the RMVC algorithm.

Table 8. The final soft set matrix representation for 12 criteria (unified threshold).

Parameter	A1	A2	A3	A4	A5	A6	A7	A8
e_1: Price	1	1	1	0	0	1	1	0
e_2: Operating cost	1	1	0	0	0	1	1	0
e_3: Resale value	0	0	0	1	1	0	0	1
e_4: Maintainability	0	0	1	0	1	1	0	1
e_5: City mileage	1	0	0	0	0	1	0	0
e_6: Highway mileage	1	0	0	0	0	1	0	0
e_7: Acceleration	0	0	1	1	1	0	0	1
e_8: Braking	0	1	1	1	0	0	1	0
e_9: Ride	0	1	1	0	1	0	1	0
e_{10}: Handling	1	1	1	1	0	0	1	0
e_{11}: Front-seat room	0	0	1	0	1	0	0	1
e_{12}: Trunk space	0	0	1	1	0	0	1	1

5.4. Application of the RMVC Algorithm and Results

With the soft set representation of the problem established in Table 8, we now apply the RMVC algorithm. The core of this method is the construction of the relational membership matrix, $\mathcal{M}_{m\times n}$.

Unlike a simple binary value, each entry in this matrix, $\Omega_{\Theta_{e_i}(A_j)}$, is a normalized score between 0 and 1. It quantifies how strongly an alternative A_j is associated with a parameter e_i, considering its relationship with all other alternatives. For alternatives that are members of the parameter set $\Phi(e_i)$, this value is 1.0. For non-members, the value is the calculated relational score.

The resulting relational membership matrix, $\mathcal{M}$, based on RMVC results, is presented in Table 9.

Table 9. The final relational membership matrix, $\mathcal{M}$.

Parameter	A1	A2	A3	A4	A5	A6	A7	A8
e_1: Price	1.00	1.00	1.00	0.18	0.13	1.00	1.00	0.11
e_2: Operating cost	1.00	1.00	0.30	0.14	0.07	1.00	1.00	0.05
e_3: Resale value	0.03	0.09	0.36	1.00	1.00	0.06	0.15	1.00
e_4: Maintainability	0.14	0.16	1.00	0.20	1.00	1.00	0.20	1.00
e_5: City mileage	1.00	0.23	0.18	0.05	0.05	1.00	0.23	0.05
e_6: Highway mileage	1.00	0.23	0.18	0.05	0.05	1.00	0.23	0.05
e_7: Acceleration	0.07	0.16	1.00	1.00	1.00	0.09	0.23	1.00
e_8: Braking	0.20	1.00	1.00	1.00	0.18	0.14	1.00	0.18
e_9: Ride	0.18	1.00	1.00	0.25	1.00	0.16	1.00	0.20
e_{10}: Handling	1.00	1.00	1.00	1.00	0.15	0.18	1.00	0.15
e_{11}: Front-seat room	0.06	0.15	1.00	0.27	1.00	0.12	0.21	1.00
e_{12}: Trunk space	0.14	0.25	1.00	1.00	0.25	0.11	1.00	1.00

The final score, $S(A_j)$, for each alternative is calculated by summing the values in its corresponding column of the matrix. A higher score indicates a more preferable alternative:

$$S(A_j) = \sum_{i=1}^{12} \Theta_{e_i}(A_j)$$

These calculations yield the final scores and the resulting RMVC ranking. To provide a comprehensive benchmark, we compare these results with two different TOPSIS rankings:

(1) A "focused TOPSIS" analysis using the exact same 12 criteria that we already calculated.

(2) The "original TOPSIS" ranking from Yoon (1987), which used all 16 criteria.

This multi-faceted comparison is presented in Table 10.

Table 10. Final comprehensive comparison: RMVC vs. focused and original TOPSIS.

Alternative	RMVC Score	RMVC Rank	Focused TOPSIS Rank (12 Criteria)	Original TOPSIS Rank (16 Criteria)
A_1 (Ford Pinto)	5.82	6	2	8
A_2 (Chevrolet Nova)	6.27	3	3	5
A_3 (Plymouth Duster)	**9.02**	**1**	5	6
A_4 (Mercury Monarch)	6.14	4	6	4
A_5 (Ford Granada)	5.86	5 (Tie)	7	2
A_6 (Cadillac Seville)	5.86	5 (Tie)	**1**	7
A_7 (Volvo 244 DL)	7.25	2	4	3
A_8 (Peugeot 504)	5.78	8	8	**1**

5.5. Discussion of Final Results

The comprehensive comparison presented in Table 10 provides a powerful validation of the RMVC method's unique approach, revealing three different optimal choices depending on the methodology and the scope of the criteria.

1. TOPSIS's sensitivity to scope: The TOPSIS method proves to be highly sensitive to the set of criteria. With all 16 criteria, the original study identifies A8 (Peugeot 504) as the best choice, likely due to its strengths in several high-weight criteria that were excluded from our 12-item set. However, when focused on our 12-criteria set, the optimal choice dramatically shifts to A6 (Cadillac Seville). This demonstrates how criterion selection and weighting can fundamentally alter the outcome of utility-based methods.
2. RMVC's robustness and consistency: On the other hand, the most compelling finding is the stability of the result produced by the RMVC method. A key strength of the proposed methodology is its consistency across varying levels of analytical scope. Preliminary analyses using a more focused 6-criteria subset, the 16-criteria set itself, and the comprehensive 12-criteria analysis presented here, all identify A3 (Plymouth Duster) as the clear and unwavering optimal choice. This remarkable stability stems from the RMVC's weight-agnostic and relational nature. The method does not seek a "utility champion" based on explicit weights but rather identifies a "consensus champion" based on overall profile balance. A3 consistently emerges as the most "well-connected" alternative that satisfies the broadest range of criteria, proving its robustness regardless of the analytical scope.

In conclusion, this final analysis powerfully illustrates the value of the RMVC method. While TOPSIS provides valuable insights based on pre-defined utility and weights, its results can be volatile. RMVC offers a more stable and robust alternative, identifying the most balanced and consensual choice by leveraging the intrinsic relationships within the data. This demonstrates its strength as a new and reliable decision-making tool.

6. Conclusions and Future Work

This paper addressed a fundamental challenge in decision-making under uncertainty: the information loss and ranking ambiguity inherent in classical soft set theory. We have put forth the Relational Membership Value Calculator (RMVC), an algorithmic framework designed to overcome these limitations by introducing a more nuanced, fine-grained approach to evaluating candidates. Our primary contributions and findings are as follows:

- We introduced a unique function, Θ, that evaluates the implicit connections between a candidate's existing attributes and those it lacks, thereby preserving critical information that is otherwise discarded.
- We provided a rigorous analysis of the algorithm's theoretical properties, establishing its polynomial-time complexity and its robustness against data sparsity.
- Through a comprehensive sensitivity analysis, we demonstrated the stability of the RMVC's output rankings under data perturbations.
- Using the Gini Index, we provided quantitative evidence that the RMVC framework achieves significantly higher granularity, leading to highly differentiated rankings with a drastically reduced likelihood of ties.

The implications of these findings are significant. The RMVC provides a more trustworthy foundation for automated and semi-automated decision-making by producing more reliable rankings with minimal ambiguity. This makes it a valuable tool for integration into next-generation AI-driven decision support and expert systems, where decision ambiguity can lead to critical failures. The provision of the algorithm as an open-source Python program further enhances its value to the scientific community by ensuring transparency and facilitating adoption.

Despite these promising results, we acknowledge several limitations that open avenues for future research. Our quantitative validation was based on illustrative case studies; future work should involve applying the RMVC to diverse, large-scale, real-world datasets to further assess its generalizability.

This work opens up several promising avenues for future investigation, which can be categorized into theoretical extensions, advanced aggregation methods, and enhancements for practical application.

A primary direction for future work involves extending the RMVC model from its current crisp framework to operate in more complex environments. Future work could focus on the following:

- Fuzzy framework: The crisp δ function could be replaced by a fuzzy counterpart (e.g., using a t-norm). The development of a Fuzzy Relational Membership Vision Clustering (F-RMVC) algorithm is a compelling next step.
- Advanced uncertainty models: Furthermore, the model can be adapted for even more sophisticated frameworks such as interval-valued fuzzy sets, intuitionistic fuzzy sets, or neutrosophic sets. Each extension would allow the algorithm to handle not just uncertainty but also concepts like indeterminacy and contradictory information, significantly broadening its applicability.

The current model uses a raw sum for score aggregation, a deliberate choice to establish a parameter-free baseline. Future work could investigate more advanced aggregation techniques to create a more flexible and adaptive model, including the following:

- Weighted aggregation: Incorporating criterion weights would allow the model to reflect the varying importance of different parameters in real-world scenarios.
- Entropy-based weighting: Information theory could be used to automatically assign higher weights to more "discriminative" criteria, providing a data-driven approach to weighting.

Future research can also focus on strengthening the model's practical and theoretical reliability through the following:

- Handling parameter redundancy: The current model assumes parameters are distinct. An advanced extension could involve an automated, in-built mechanism to account for highly correlated or redundant criteria, for example, by down-weighting the contribution of similar parameters.
- Formalizing stability guarantees: While traditional error bounds are not directly applicable, the concept of robustness can be formalized. Future work could involve defining a distance metric between rankings (e.g., Kendall's Tau) and then deriving probabilistic bounds on the stability of the output, further solidifying the model's theoretical foundations.

An important objective for future work is to perform the formal comparative analysis theorized in this paper. This would involve using metrics to rigorously validate the enhanced granularity of the RMVC method, including the following:

- Gini Index: A lower Gini Index for RMVC scores compared to traditional methods would provide quantitative proof of a more distributed score space and a reduced probability of ties.
- Kullback–Leibler (KL) divergence: A high KL divergence value when comparing distributions would further support the argument that the RMVC provides a richer and more informative ranking.

Such an investigation would provide a standardized framework for comparing the resolution power of various decision-making algorithms. In conclusion, this work contributes a validated, transparent, and robust algorithmic tool to the field, paving the way for more precise and reliable data-driven decision-making in complex environments.

Author Contributions: Conceptualization, A.D. and B.C.; methodology, A.D. and F.O.E.; software, A.D.; validation, A.D. and F.O.E.; formal analysis, A.D.; investigation, A.D. and F.O.E.; resources, A.D. and F.O.E.; data curation, A.D. and F.O.E.; writing—original draft preparation, A.D.; writing—review and editing, A.D.; visualization, A.D.; supervision, B.C.; funding acquisition, A.D., F.O.E., and B.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The open-source Python code for the RMVC program is available at https://github.com/DrDayioglu/RMVC.git, accessed on 30 July 2025.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Molodtsov, D. Soft set theory first results. *Comput. Math. Appl.* **1999**, *37*, 19–31. [CrossRef]
2. Maji, P.K.; Biswas, R.; Roy, A.R. Soft set theory. *Comput. Math. Appl.* **2002**, *45*, 555–562. [CrossRef]
3. Maji, P.K.; Roy, A.R.; Biswas, R. An application of soft sets in a decision making problem. *Comput. Math. Appl.* **2003**, *44*, 1077–1083. [CrossRef]
4. Chen, D.; Tsang, E.C.C.; Yeung, D.S.; Wang, X. The parameterization reduction of soft sets and its applications. *Comput. Math. Appl.* **2005**, *49*, 757–763. [CrossRef]
5. Ali, M.I.; Feng, F.; Liu, X.; Min, W.K.; Shabir, M. On some new operations in soft set theory. *Comput. Math. Appl.* **2007**, *57*, 1547–1553. [CrossRef]
6. Kong, Z.; Gao, L.; Wang, L.; Li, S. The normal parameter reduction of soft sets and its algorithm. *Comput. Math. Appl.* **2008**, *56*, 3029–3037. [CrossRef]
7. Cagman, N.; Enginoglu, S. Soft matrix theory and its decision making. *Comput. Math. Appl.* **2010**, *59*, 3308–3314. [CrossRef]
8. Peng, X.; Yang, Y. Algorithms for interval-valued fuzzy soft sets in stochastic multi-criteria decision making based on regret theory and prospect theory with combined weight. *Appl. Soft Comput.* **2017**, *54*, 415–430. [CrossRef]
9. Peng, X.; Liu, C. Algorithms for neutrosophic soft decision making based on EDAS, new similarity measure and level soft set. *J. Intell. Fuzzy Syst.* **2019**, *32*, 955–968. [CrossRef]
10. Smarandache, F. Extension of soft set to hypersoft set, and then to plithogenic hypersoft set. *Neutrosophic Sets Syst.* **2018**, *22*, 168–170.
11. Ahmad, B.; Kharal, A. On fuzzy soft sets. *Adv. Fuzzy Syst.* **2009**, *2009*, 586507. [CrossRef]
12. Akram, M.; Luqman, A. *Fuzzy Hypergraphs and Related Extensions*; Studies in Fuzziness and Soft Computing; Springer: Singapore, 2020; Volume 390.
13. Demirtas, N.; Dalkilic, O. An application of fuzzy soft relation in decision making. *J. New Theory* **2020**, *30*, 98–107.
14. Dalkilic, O. A novel approach to soft set theory in decision-making under uncertainty: New definitions and applications. *Int. J. Comput. Math.* **2021**, *98*, 1935–1945. [CrossRef]
15. Dalkilic, O. On topological structures of fuzzy parametrized soft sets. *J. Math.* **2021**, *2021*, 3942708.
16. Dalkilic, O. Relations on neutrosophic soft set and their application in decision making. *J. Appl. Math. Comput.* **2021**, *67*, 257–273. [CrossRef]
17. Dalkilic, O.; Demirtas, N. VFP-soft sets and its application on decision making problems. *J. Polytech.* **2021**, *24*, 1581–1589.
18. Nawaz, H.S.; Akram, M. Granulation of protein-protein interaction networks in Pythagorean fuzzy soft environment. *J. Appl. Math. Comput.* **2021**, *67*, 293–320. [CrossRef]
19. Siddique, I.; Ahmad, T.; Awrejcewicz, J. A novel approach to study the effect of temperature and concentration on dynamic viscosity of MgO-Ag/water hybrid nanofluid using artificial neural network. *Case Stud. Therm. Eng.* **2021**, *26*, 101055.
20. Zulqarnain, R.M.; Xin, X.L.; Garg, H.; Khan, W.A. Aggregation operators of pythagorean fuzzy soft sets with their application for green supplier chain management. *J. Intell. Fuzzy Syst.* **2021**, *40*, 5545–5563. [CrossRef]

21. Zulqarnain, R.M.; Xin, X.L.; Saeed, M.; Ahmad, N.; Dayan, F.; Ahmad, B. Some fundamental operations on interval valued neutrosophic hypersoft set with their properties. *Neutrosophic Sets Syst.* **2021**, *40*, 134–148.
22. Zulqarnain, R.M.; Xin, X.L.; Saqlain, M.; Khan, W.A. TOPSIS method based on correlation coefficient under pythagorean fuzzy soft environment and its application towards green supply chain management. *Sustainability* **2021**, *13*, 1642. [CrossRef]
23. Zulqarnain, R.M.; Xin, X.L.; Saeed, M.; Smarandache, F.; Ahmad, N. Generalized aggregate operators on neutrosophic hypersoft set. *Neutrosophic Sets Syst.* **2021**, *36*, 271–281.
24. Sezgin, A.; Atagun, A.O.; Cagan, N. A complete study on and-product of soft sets. *SIGMA* **2025**, *43*, 1–14. [CrossRef]
25. Xia, S.; Chen, L.; Yang, H. A soft set based approach for the decision-making problem with heterogeneous information. *AIMS Math.* **2022**, *7*, 20420–20440. [CrossRef]
26. Cetkin, V.; Aygunoglu, A.; Aygun, H. A new approach in handling soft decision making problems. *J. Nonlinear Sci. Appl.* **2016**, *9*, 231–239. [CrossRef]
27. Yoon, K. A Reconciliation Among Discrete Compromise Solutions. *J. Oper. Res. Soc.* **1987**, *38*, 277–286. [CrossRef]
28. Fleischer, G.A. *Engineering Economy*; PWS Engineering: Boston, MA, USA, 1984.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.