Unsupervised Fuzzy Multi-Objective Optimization for Complex Decision-Making Problems

1st Diyar Altinses, 2nd Sofiene Lassoued, 3rd David O. Salazar Torres, 4th Stefan Lier, 5th Andreas Schwung
Institute for Sustainable Technologies and Integrated Production Systems
South Westphalia University of Applied Sciences
Soest, Germany
altinses.diyar, lassoued.sofiene, salazar.davidorlando, lier.stefan, schwung.andreas@fh-swf.de

***Abstract*—Many real-world optimization problems involve multiple competing objectives, where the exact weighting is often difficult to define. Traditional fuzzy logic approaches for decision-making, while useful for handling uncertainty, require evaluating all combinations to find the optimal solution. For instance, in logistical scenarios where the optimal location must be determined, fuzzy logic would require evaluating every point on the map, which is time-consuming and computationally expensive. This paper presents an unsupervised neuro-fuzzy optimization model that addresses these problems. Our approach combines a multimodal neural network with differentiable fuzzy objectives for efficient optimization. The model provides robust solutions across various applications, including location selection, network infrastructure, and energy distribution. By leveraging differentiable fuzzy objectives, it can handle complex multi-objective tasks while ensuring high performance and scalability. This method significantly improves traditional optimization techniques and is highly applicable in dynamic, real-time environments.**

***Index Terms*—Neuro-Fuzzy Optimization, Unsupervised Learning, Multi-Objective Optimization, Differentiable Fuzzy Logic Criterion**

I. Introduction

Many real-world optimization problems involve multiple, often conflicting objectives, which are crucial in various application domains. In logistics, for example, companies must simultaneously minimize delivery time, reduce costs, and implement sustainable transportation solutions [1]. Similarly, in the energy sector, the challenge lies in balancing maximum efficiency, grid stability, and cost optimization [2]. In medicine, the development of optimal treatment strategies is essential, considering multiple patient factors [3].

Traditional methods for solving such problems often rely on heuristic rules or manual weighting of individual criteria. However, these approaches are typically suboptimal, as they do not systematically evaluate the entire solution space and fail to fully capture potential trade-offs [4]. Additionally, the problem becomes increasingly complex as more criteria and influencing factors are considered. Another approach to addressing such multi-criteria optimization problems is the use of fuzzy logic. This method allows for modeling uncertainties and gradual transitions between different decision options rather than defining rigid thresholds [5]. However, a significant issue with fuzzy logic methods in decision-making is that they must first evaluate all possible combinations of input values and their impact on target variables to identify an optimal solution. This results in substantial computational complexity and limits the efficiency of the method in practical applications.

An efficient and innovative solution is to use multimodal machine learning, especially unsupervised learning, to process the entire data at once rather than point by point. Unlike supervised learning, which relies on labeled data, unsupervised learning enables the discovery of patterns and structures in data without explicit labels or target values [6]. An additional level of complexity of multi-objective learning arises when dealing with different data inputs or modalities. In such cases, a multimodal approach is required to effectively integrate and utilize the information from the different sources. Multimodal learning allows the complementary information from different modalities to develop a more comprehensive understanding of the problem [7].

In this paper, we propose a generic method for unsupervised fuzzy multi-objective optimization. We employ fuzzy logic as the optimization criterion and optimize neural networks using gradient descent. Initially, the data is represented as a low-dimensional vector using neural models, then fused and fed into the decision-making model. We have developed a differentiable fuzzy objective function that is directly optimizable, from decision-making to evaluation, using fuzzy logic. This approach balances competing objectives in an unsupervised manner. Our key contributions can be summarized as follows:

(i) We introduce a novel framework that seamlessly integrates unsupervised learning with fuzzy logic to tackle complex optimization challenges. This model is designed to be adaptable across different domains and scalable to handle large datasets with high-dimensional input spaces.
(ii) We design and implement fuzzy-based cost functions that can be directly optimized through gradient-based learning techniques. These functions enable neural networks to effectively learn from uncertain and imprecise data, enhancing decision-making in multi-objective scenarios.
(iii) To validate our approach, we conduct experiments of a real-world use case. This includes optimizing supply chain logistics by identifying positions for reduced costs and improved efficiency.

The paper begins by reviewing multi-objective optimization, fuzzy learning, and multimodal prediction in Section 2. Section 3 then presents the gradient flow theory and machine learning architecture. Experiments and results follow in Section 4, evaluating the approach across diverse datasets. Finally, the study concludes by summarizing key findings and suggesting future applications in Section 5.

II. Related Work

This section addresses related work of three critical areas that are central to the focus of this paper: Multimodal Learning, Multi-Objective Optimization, and Neural Fuzzy.

A. Multimodal Learning

Multimodal learning focuses on integrating different data types, such as spatial and numerical information, to enhance predictive accuracy and decision-making processes. Various approaches have been developed to address feature fusion and representation learning in multi-objective problems.

One popular technique involves early fusion, where raw data from multiple modalities are combined at the input level before feature extraction [8]. While this approach ensures that all information is available to the model from the beginning, it can suffer from high dimensionality and redundant information [9]. Alternatively, late fusion methods process each modality separately before merging their high-level representations, allowing for more flexible feature extraction [10] [11].

Recent studies have explored attention-based fusion mechanisms, which dynamically weight different modalities based on their relevance to the task [12]. These models have been particularly successful in applications such as medical diagnosis and autonomous systems, where multiple data sources contribute to decision-making [13]. Furthermore, deep generative models like variational autoencoders and transformer-based architectures have demonstrated superior performance in learning complex, multimodal feature spaces [14].

In the context of multi-objective optimization, leveraging multimodal learning allows for more comprehensive decision modeling. Studies have shown that combining heterogeneous data sources significantly improves optimization accuracy and generalization across different application domains. However, challenges remain in effectively balancing the contributions of different modalities and unsupervised multimodal decision-making.

B. Multi-Objective Optimization

Multi-objective optimization aims to simultaneously optimize multiple conflicting objectives, requiring advanced strategies to achieve balanced solutions [15]. Classical methods such as weighted sum approaches, Non-dominated Sorting Genetic Algorithm II (NSGA-II), and Pareto-based optimization have been widely used in various domains [16]. However, these methods often struggle with nonlinearity, uncertainty, and dynamic changes in real-world problems [4].

Traditional optimization techniques rely on predefined weightings or explicit trade-off curves, which can be rigid and impractical, specifically with multiple complex conflicting objectives [17]. Recent advancements have introduced adaptive and learning-based approaches, where neural networks and reinforcement learning techniques dynamically adjust optimization parameters [18] [19]. However, to our knowledge, no prior work has employed differentiable fuzzy logic criteria as optimization objectives for gradient descent.

C. Fuzzy Neural Networks

Fuzzy Neural Networks (FNNs) integrate the learning capabilities of neural networks with the reasoning approach of fuzzy logic, enabling the modeling of complex systems under uncertainty. This synergy enhances system performance in various applications, including pattern recognition, control systems, and data classification.

Recent work by Das et al. explored the enhancement of deep learning performance through the integration of fuzzy systems, classifying techniques based on their combination methods and examining real-life applications of these models [20]. Fan revisited FNNs by introducing the concept of generalized Hamming distance, providing a novel framework to reinterpret neural network techniques such as batch normalization and ReLU within the context of fuzzy logic [21]. Price et al. proposed the incorporation of fuzzy layers into deep learning architectures, leveraging fuzzy aggregation methods like the Choquet and Sugeno integrals to enhance performance in tasks such as semantic segmentation [22]. Kulkarni introduced a novel framework called Fuzzy Convolution Neural Network (FCNN) tailored for tabular data classification, mapping feature values to fuzzy memberships and converting them into images for CNN training, achieving competitive performance compared to existing methods [23].

Collectively, these studies highlight the adaptability and effectiveness of FNNs in tackling complex and uncertain systems across various applications. While previous research has primarily focused on integrating classical neural networks with traditional fuzzy logic within the network layers, our approach takes a different direction by incorporating fuzzy logic into the objective function, leading to the development of a differentiable fuzzy loss.

III. Unsupervised Fuzzy Multi-Objective Optimization

In this section, we begin by defining the challenges related to unsupervised multi-objective optimization using fuzzy criteria. After outlining the challenges, we present our multimodal neural architecture that fuses varying heterogeneous data and makes a decision. Then we introduce our fuzzy objective function and the corresponding extension to retain the gradients for the optimization process. In Figure 1 we present the overall setup.

A. Problem defintion

Given a dataset $\mathcal{D}$ consisting of N distinct modalities, let $\{\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_N\}$ represent the modalities, where each $\mathbf{X}_i$ is a matrix, i.e., $\mathbf{X}_i \in \mathbb{R}^{m_i \times n_i}$, corresponding to a potential decision map, such as a communication connection map. The

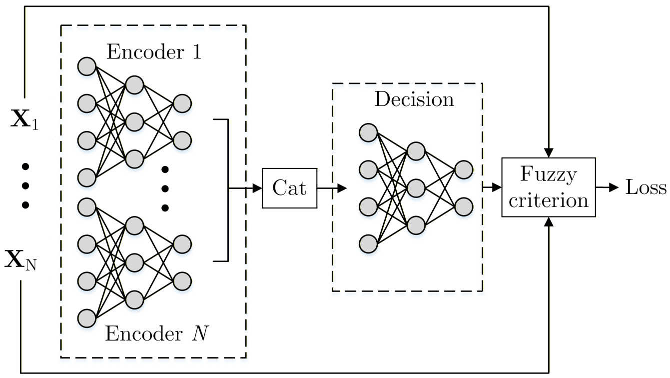


Fig. 1: Overview of the proposed multimodal neural architecture for unsupervised multi-objective optimization using fuzzy criteria.

goal is to optimize a function $f : \mathbb{R}^p \to \mathbb{R}^q$, parameterized by $\boldsymbol{\theta}$, based on a fuzzy objective function $\mathcal{L}_f(\mathbf{X}, \boldsymbol{\theta})$, where the input to the fuzzy objective function is the output of the parameterized function $f(\mathbf{X}, \boldsymbol{\theta})$, which is applied to the input matrices $\mathbf{X} \in \mathcal{D}$. Formally, the fuzzy objective function can be defined as:

$$\min_{\boldsymbol{\theta}} \mathcal{L}_f(\mathbf{X}, \boldsymbol{\theta}) = \min_{\boldsymbol{\theta}} \sum_{i=1}^{N} \mathcal{F}(f(\mathbf{X}_i, \boldsymbol{\theta}), \mathbf{X}_i), \tag{1}$$

where the function $\mathcal{F}(\cdot)$ is a fuzzy aggregation operator, such as the Sugeno or Choquet integral, that computes the fuzzy score for the decision at each modality. The goal is to find the optimal parameter $\boldsymbol{\theta}^*$ that minimizes the fuzzy loss function $\mathcal{L}_f$, which leads to the optimal position in the decision space.

B. Multimodal neural network architecture

The proposed neural network architecture is designed to handle multimodal data by incorporating N encoders, each corresponding to a distinct modality. Let the set of modalities be denoted as $\{\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_N\}$, where each $\mathbf{X}_i \in \mathbb{R}^{m_i \times n_i}$ represents the input data for modality i, with dimensions $m_i \times n_i$. The goal is to unify these modalities by transforming them into a shared representation space.

Each modality $\mathbf{X}_i$ is passed through a modality-specific encoder, $E_i : \mathbb{R}^{m_i \times n_i} \to \mathbb{R}^d$, which maps the input data $\mathbf{X}_i$ into a common latent space of dimensionality d:

$$\mathbf{z}_i = E_i(\mathbf{x}_i), \quad \forall i \in \{1, 2, \ldots, N\}. \tag{2}$$

This transformation ensures that the data from all modalities are represented in a unified structure and size, making them suitable for subsequent fusion and decision-making. The concatenated multimodal representation is then given by:

$$\mathbf{z}_{\text{concat}} = [\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_N] \in \mathbb{R}^{N \cdot d}. \tag{3}$$

This concatenated vector $\mathbf{z}_{\text{concat}}$ is passed as input to a decision-making network $D : \mathbb{R}^{N \cdot d} \to \mathbb{R}^2$, which is designed to output the optimal decision indices, x_{opt} and y_{opt}, corresponding to the optimal position in the input data. The decision-making process is governed by the function:

$$\mathbf{y} = D(\mathbf{z}_{\text{concat}}) = [x_{\text{opt}}, y_{\text{opt}}]^T \in \mathbb{R}^2. \tag{4}$$

To optimize the parameters of all the submodels, we define a fuzzy objective function which is optimized by gradient descent.

C. Fuzzy objective function

The fuzzy objective function utilizes fuzzy logic to map inputs to outputs through a series of transformations governed by fuzzy rules. Let the fuzzy inputs $\mathbf{x} = \{x_1, x_2, \ldots, x_n\} \in \mathbb{R}^n$ be the variables, each described by a fuzzy membership function $\mu_A(x_i)$, where $A \in \mathcal{A}$ is a fuzzy set, and x_i is the i-th input. The process is based on the work [5] and compactly described by the following stages:

(i) Fuzzification: Convert the crisp inputs x_i into fuzzy values by applying the corresponding membership functions $\mu_A(x_i)$:

$$\mu_A(x_i) = \frac{1}{1 + e^{-\alpha_i (x_i - \beta_i)}},$$

where α_i and β_i are parameters governing the steepness and center of the membership function.

(ii) Rule Evaluation: The fuzzy inference system consists of r fuzzy IF-THEN rules of the form:

$$\text{IF } x_1 \text{ is } A_1 \text{ AND } x_2 \text{ is } A_2 \text{ THEN } y \text{ is } B,$$

where $A_1, A_2 \in \mathcal{A}$ are fuzzy sets, and $B \in \mathcal{B}$ is the output fuzzy set. The fuzzy firing strength of rule i is given by the minimum of the membership values for the input variables:

$$w_i = \min\left(\mu_{A_1}(x_1), \mu_{A_2}(x_2)\right).$$

(iii) Aggregation: The results of the individual rules are aggregated into a single fuzzy output set $\mathcal{B}$. This is typically done by summing the weighted fuzzy outputs of all rules:

$$\mathcal{B} = \bigcup_{i=1}^{r} w_i B_i,$$

where B_i represents the fuzzy output set of rule i.

(iv) Defuzzification: The final crisp output y_{defuzz} is obtained by applying a defuzzification operator, such as the centroid method:

$$y_{\text{defuzz}} = \frac{\int_{\mathcal{B}} y \cdot \mu_{\mathcal{B}}(y)\, dy}{\int_{\mathcal{B}} \mu_{\mathcal{B}}(y)\, dy},$$

where $\mu_{\mathcal{B}}(y)$ is the membership function of the aggregated fuzzy output set.

Thus, the fuzzy inference process transforms fuzzy inputs through rule-based evaluation, aggregation, and defuzzification to yield a crisp output.

D. Gradients using bilinear interpolation

A particular challenge arises when gradient information must be preserved during the indexing process using the float values of the neural network to select the point in the input data and extract the positional information from the decision. Without retaining the gradient, this model will lose the end-to-end differentiability by integrating the fuzzy loss function into the neural network, and the optimization or learning process will be affected. Therefore, we propose to use bilinear interpolation, as it allows for the extraction of values from the grid while ensuring the preservation of both the values and their gradients in the continuous domain.

Given a 2D grid of data points $f(x_i, y_j)$, where x_i and y_j are integer grid indices, and x and y are the continuous query coordinates between the grid points, we aim to interpolate the value $f(x, y)$ at the floating-point coordinates (x, y). Bilinear interpolation is a linear interpolation applied twice, first in the x-direction and then in the y-direction. For a given point (x, y), we need to find the four surrounding grid points:

$$(x_0, y_0), \quad (x_1, y_0), \quad (x_0, y_1), \quad (x_1, y_1) \tag{5}$$

where $x_0 = \lfloor x \rfloor$, $x_1 = x_0 + 1$, $y_0 = \lfloor y \rfloor$, and $y_1 = y_0 + 1$. The bilinear interpolation for $f(x, y)$ is then:

$$\begin{aligned} f(x,y) = &(1-dx)(1-dy)f(x_0,y_0) \\ &+ dx(1-dy)f(x_1,y_0) \\ &+ (1-dx)dyf(x_0,y_1) \\ &+ dxdyf(x_1,y_1) \end{aligned} \tag{6}$$

where $dx = x - x_0$ is the fractional part of x and $dy = y - y_0$ is the fractional part of y.

The key aspect of bilinear interpolation is that it ensures continuous changes in the output values as the input coordinates (x, y) vary. Specifically, the gradient with respect to x and y is continuous, ensuring smooth transitions between neighboring grid points. To demonstrate the gradient preservation, we compute the partial derivatives of $f(x, y)$ with respect to both x and y. These partial derivatives express how the interpolated value changes with respect to small changes in the input coordinates.

$$\begin{aligned} \frac{\partial f}{\partial x} = &(1-dy)\left(f(x_1,y_0) - f(x_0,y_0)\right) \\ &+ dy\left(f(x_1,y_1) - f(x_0,y_1)\right) \end{aligned} \tag{7}$$

$$\begin{aligned} \frac{\partial f}{\partial y} = &(1-dx)\left(f(x_0,y_1) - f(x_0,y_0)\right) \\ &+ dx\left(f(x_1,y_1) - f(x_1,y_0)\right) \end{aligned} \tag{8}$$

These partial derivatives are computed directly from the neighboring points and thus allow for the preservation of the gradient as the coordinates (x, y) change.

IV. Evaluation

In this section, we introduce the dataset and outline the experimental setup used to obtain the corresponding results. We first outline the data collection process and key characteristics of the dataset. The experimental framework is structured into two main components: a description of the model architecture, including the optimization strategy for tuning its parameters, and second, the formal definition of the fuzzy cost function. Finally, we present the corresponding results of our approach.

A. Dataset

The dataset $\mathcal{D}$ is a synthetic logistics dataset designed for the optimal placement of microhubs in a multimodal delivery scenario, where drones are deployed from strategically located hubs to serve customer demands. The dataset consists of three primary modalities: wind conditions, network communication quality, and potential customer distribution.

Drone aerodynamics and energy use are highly wind-dependent. Headwinds drain batteries and delay deliveries, while crosswinds destabilize flights. Microhubs require locations with favorable wind profiles to ensure operational reliability and energy efficiency. Additionally, uninterrupted communication is critical for navigation and coordination. Weak signals in urban or remote areas disrupt telemetry, necessitating microhubs in zones with robust coverage to maintain control and comply with safety regulations. Lastly, demand concentration determines optimal hub placement. Poor positioning increases delivery costs, while over-clustering wastes resources. Spatial analysis of orders balances workload and ensures scalable, cost-effective service. These three interdependent factors define microhub placement strategies.

(i) Wind Field Representation: The wind data is modeled as a continuous two-dimensional gradient field $\mathbf{W}(\mathbf{x}) \in \mathbb{R}^2$, where each spatial location $\mathbf{x} = (x_1, x_2)$ is associated with a vector $\mathbf{w} = (\theta, m)$, representing the wind direction $\theta \in [0, 2\pi)$ and magnitude $m \in \mathbb{R}_{\geq 0}$.

(ii) Network Communication Quality: The network quality is encoded as a procedural texture primitive, specifically a gradient noise function $\mathcal{N}(\mathbf{x})$, represented as a three-dimensional matrix $\mathbf{C} \in \mathbb{R}^{H \times W \times D}$, where H, W, and D define the spatial and spectral resolution of the communication landscape.

(iii) Customer Distribution: The potential customer locations are provided as a discrete spatial matrix $\mathbf{P} \in \mathbb{R}^{N \times 2}$, where each row corresponds to a customer position $\mathbf{p}_i = (p_{i,1}, p_{i,2})$ in the two-dimensional search space.

The dataset is generated following the methodology introduced in [24] and consists of $K = 500$ distinct scenarios, each characterized by a unique instantiation of the wind field, communication quality map, and customer distribution. The dataset structure is designed to be extendable, allowing for an arbitrary number of additional scenarios $K' \geq K$ with varying environmental conditions and demand patterns.

B. Experimental Setup

The experimental setup consists of two primary components: model architecture and parameter optimization and the definition of the fuzzy cost function. The model architecture and parameter optimization section provides a detailed description of the proposed model, including its structural design, the employed neural network layers, and the parameterization strategy. The definition of the fuzzy cost function formalizes the construction of the fuzzy cost function, detailing the membership functions, rule-based inference mechanism, and defuzzification process. Together, these components define the overall experimental framework.

1) Model architecture: Due to the heterogeneous nature of the input data, we employ distinct encoder architectures tailored to the characteristics of each modality. High-dimensional data is processed using convolutional neural networks, while low-dimensional data is flattened and transformed via fully connected layers.

(i) For the wind modality, we utilize a convolutional architecture with a layer configuration of 4-16-16-16, where each value denotes the number of feature channels at each stage of the network. The convolutional layers employ a kernel size of 10 and a stride of 2, with ReLU activations applied between all layers.
(ii) Similarly, for the communication quality map, we use a 2D convolutional encoder with the configuration 1-16-16-16, maintaining the same kernel size of 10 and stride of 2, with ReLU activations at each stage.
(iii) For the customer positions, we apply a fully connected architecture. Specifically, we flatten the customer positions into a feature vector and process them through three dense layers with dimensions 20-20-20, incorporating ReLU activations between all layers.

The encoded representations of all input modalities are concatenated, forming a feature vector of size $\mathbb{R}^{84}$. This vector is subsequently passed through a decision model with a fully connected architecture structured as 84-84-2, again utilizing ReLU activations between layers.

The entire model is trained for 100 epochs with a batch size of 16 using the Adam optimizer with a learning rate of 10^{-4}. The optimization is performed using the Mean Squared Error loss as the objective function.

2) Fuzzy criterion: To evaluate the quality of microhub placement within the multimodal delivery scenario, we normalize corresponding properties between 0 and 100 and then employ the fuzzy logic-based decision-making system. This system defines membership functions for input variables, establishes inference rules, and applies a structured defuzzification process to derive the final placement score.

First, each input and the corresponding target score are classified into three linguistic categories: low, mid, and high. All membership functions are modeled as Gaussian functions parameterized by a central value c and a width parameter σ:

$$\mu_{\text{low}}(x) = \exp\left(-\frac{(x-0)^2}{2\cdot 30^2}\right), \tag{9}$$

$$\mu_{\text{mid}}(x) = \exp\left(-\frac{(x-50)^2}{2\cdot 30^2}\right), \tag{10}$$

$$\mu_{\text{high}}(x) = \exp\left(-\frac{(x-100)^2}{2\cdot 30^2}\right). \tag{11}$$

The fuzzy inference system applies the following rule-based decision logic to determine the placement score:

(i) If wind is low OR communication is high OR distance is low, then score is low.
(ii) If wind is high OR communication is low OR distance is high, then score is high.

Each rule is evaluated based on the corresponding fuzzy membership degrees, and the results are aggregated using the product implication function:

$$\mu_{\text{out}}(S) = \prod_{i=1}^{R} \mu_{\text{rule}_i}(S) \tag{12}$$

where R is the number of active rules. The overall output membership function is computed using the conjunction operator and the final placement score is derived using the center of gravity (COG) method:

$$S^* = \frac{\int S\cdot \mu_{\text{out}}(S)dS}{\int \mu_{\text{out}}(S)dS} \tag{13}$$

ensuring a continuous and interpretable decision output.

C. Results

First, we present the training performance of the model in Figure 2, illustrating its convergence behavior over multiple runs. To ensure statistical robustness, we conducted 10 independent trials, each initialized with different random seeds.

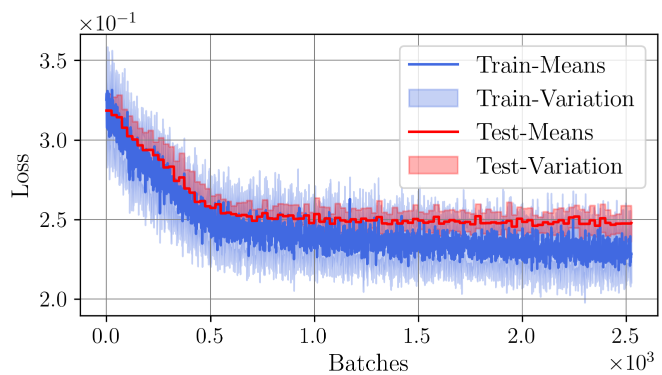
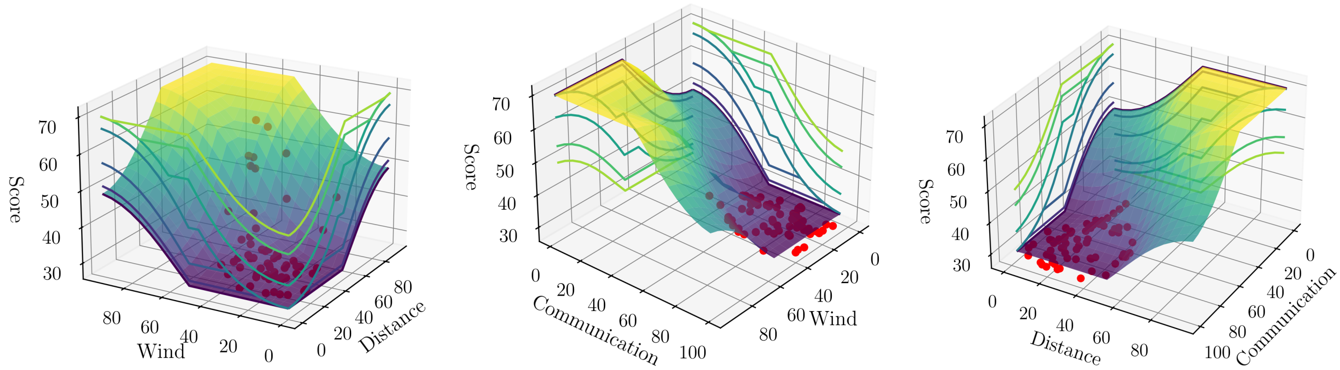


Fig. 2: Training performance of the model over 10 trials, showing the mean and standard deviation throughout the optimization process.

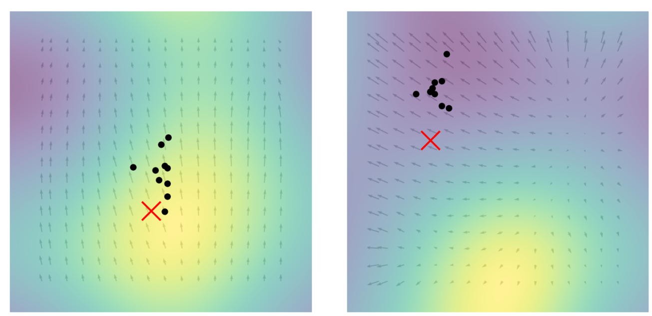
The results indicate a slight degree of overfitting as well as variations in the training process, which can be attributed to the characteristics of the fuzzy criterion surface. Specifically, the degree of fluctuation observed during training is influenced by the slope of this surface. A steeper gradient in the fuzzy cost function leads to stronger oscillations in the optimization process as the model experiences more abrupt changes in the loss landscape. Conversely, if the fuzzy surface is designed with milder slopes, the fluctuations would be significantly reduced, contingent on the learning rate used during training. This relationship between surface geometry and training stability suggests that the tuning of fuzzy membership functions controls the convergence.



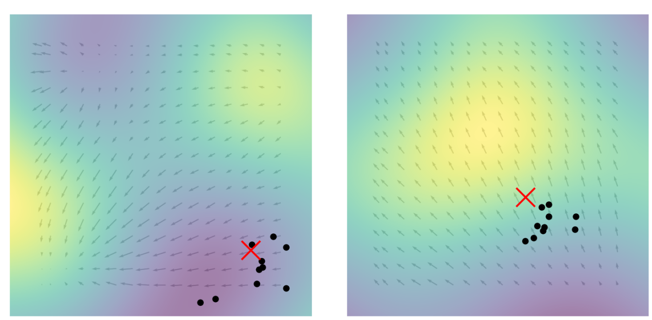
(a) Communication constant (b) Distance constant (c) Wind constant

Fig. 3: Training dynamics of fuzzy criterion optimization: Three 3D surface plots showing the functional relationship between input variables and model output. For each plot, two input variables are varied while the third is held constant. The red points on the surface represent every 50th score during the optimization process, showing that the score is gradually moving towards the minimum. The surface demonstrates how the neuro-fuzzy system balances competing objectives across different regions of the input space.

This relationship between the fuzzy criterion topology and training dynamics provides valuable insights into the interpretability of the optimization process. By carefully adjusting the smoothness of the fuzzy decision boundaries, we can enhance training stability, mitigate overfitting, and improve the overall effectiveness of the optimization procedure. This, in turn, allows for more controlled and fine-tuned parameter updates. Additionally, we can observe how the network is optimizing on the surface since we can plot the scores of the network output on the fuzzy surface as presented in Figure 3. Due to the inherent challenges of visualizing four-dimensional space, we employ a systematic decomposition: three distinct 3-D plots are generated, each illustrating the loss landscape while holding one key variable constant. This representation not only reveals how the optimizer navigates the fuzzy criterion surface but also highlights regions where gradient instability or local minima may occur, providing actionable feedback for architectural improvements.



(a) Sample 1 (b) Sample 2



(c) Sample 3 (d) Sample 4

Fig. 4: Model output predictions for 4 representative deployment scenarios. Each subplot visualizes a dual-layer representation: the heatmap background indicates network quality (warmer colors = stronger connectivity) and gradient arrows show wind patterns (arrow direction = prevailing wind, length = speed). Customer demand points are marked as black dots, with the model's microhub placement shown as a red cross.

Finally, we present the output predictions on the dataset to assess whether the model has correctly applied and optimized the inference rules. In Figure 4, we showcase four sample scenarios from the dataset, where the network quality and wind maps are displayed as background layers, with customer locations marked as black dots. The red cross symbol denotes the proposed microhub position determined by the model.

The objective of the model was to place the microhub at a location where communication quality is high, wind magnitude is low, and the distance to all customers is minimized. Any deviation from this optimal placement results in a higher loss, which the model aims to minimize. As shown in Figure 4, the model successfully achieves this objective. There is potential to refine the rule set by either adding new rules or adjusting the weight of existing ones. For example, if the distance to customers has a greater impact than other factors, the rules could be modified to reflect this priority more strongly. In conclusion, these results highlight the model's ability to effectively optimize the microhub placement while also offering interpretability.

V. Conclusion

In this paper, we introduced a generic unsupervised neuro-fuzzy optimization model designed to address key challenges in multi-objective optimization. By integrating differentiable fuzzy objectives, our approach provides a flexible and interpretable framework capable of handling complex, real-world optimization tasks. The model's effectiveness was validated across diverse applications, demonstrating consistent performance in balancing competing objectives while maintaining computational efficiency. Notably, our method excels in unsupervised settings where traditional optimization techniques struggle, offering a robust alternative for scenarios with complex optimization targets. The results highlight the model's ability to adapt to varying problem structures, making it a versatile tool for a wide range of optimization tasks.

Future work includes scaling the model to handle larger, high-dimensional, multi-objective problems; extending it to dynamic environments for real-time optimization; and developing unsupervised Pareto-optimized fuzzy networks to enhance optimization efficiency. Additionally, incorporating reinforcement learning could enable adaptive optimization, allowing the model to improve dynamically over time. Furthermore, the observed overfitting patterns imply that additional regularization techniques may be beneficial when dealing with particularly complex fuzzy optimization landscapes.

Funding

The research project was funded by "The Ministry of the Environment, Nature Conservation and Transport of the State of North Rhine-Westphalia" in Germany and co-financed by the European Union for the research project "SIDDA - Sustainable Intermodal Drone Delivery Airline" with grant number IN-ML-1-013b.

References

[1] H. Zhao, R. He, and J. Su, "Multi-objective optimization of traffic signal timing using non-dominated sorting artificial bee colony algorithm for unsaturated intersections," *Archives of Transport*, vol. 46, 2018.

[2] H. Liu, Y. Li, Z. Duan, and C. Chen, "A review on multi-objective optimization framework in wind energy forecasting techniques and applications," *Energy Conversion and Management*, vol. 224, p. 113324, 2020.

[3] G. Murugesan, T. I. Ahmed, J. Bhola, M. Shabaz, J. Singla, M. Rakhra, S. More, and I. A. Samori, "Fuzzy logic-based systems for the diagnosis of chronic kidney disease," *BioMed Research International*, vol. 2022, 2022.

[4] R. T. Marler and J. S. Arora, "Survey of multi-objective optimization methods for engineering," *Structural and multidisciplinary optimization*, vol. 26, pp. 369–395, 2004.

[5] D. Altinses, D. O. Salazar Torres, M. Schwung, S. Lier, and A. Schwung, "Optimizing drone logistics: A scoring algorithm for enhanced decision making across diverse domains in drone airlines," *Drones*, vol. 8, no. 7, p. 307, 2024.

[6] D. Altinses and A. Schwung, "Multimodal synthetic dataset balancing: a framework for realistic and balanced training data generation in industrial settings," in *IECON 2023-49th*. IEEE, 2023, pp. 1–7.

[7] D. Altinses, D. O. Salazar Torres, S. Lier, and A. Schwung, "Neural data fusion enhanced pd control for precision drone landing in synthetic environments," in *2025 IEEE International Conference on Mechatronics*. IEEE, 2025.

[8] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, A. Y. Ng *et al.*, "Multimodal deep learning." in *ICML*, vol. 11, 2011, pp. 689–696.

[9] T. Baltrušaitis, C. Ahuja, and L.-P. Morency, "Multimodal machine learning: A survey and taxonomy," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 2, pp. 423–443, 2018.

[10] D. Ramachandram and G. W. Taylor, "Deep multimodal learning: A survey on recent advances and trends," *IEEE signal processing magazine*, vol. 34, no. 6, pp. 96–108, 2017.

[11] D. Altinses and A. Schwung, "Deep multimodal fusion with corrupted spatio-temporal data using fuzzy regularization," in *IECON 2023-49th Annual Conference of the IEEE Industrial Electronics Society*. IEEE, 2023, pp. 1–7.

[12] Y.-H. H. Tsai, S. Bai, P. P. Liang, J. Z. Kolter, L.-P. Morency, and R. Salakhutdinov, "Multimodal transformer for unaligned multimodal language sequences," in *Proceedings of the conference. Association for computational linguistics. Meeting*, vol. 2019. NIH Public Access, 2019, p. 6558.

[13] F. Behrad and M. S. Abadeh, "An overview of deep learning methods for multimodal medical data mining," *Expert Systems with Applications*, vol. 200, p. 117006, 2022.

[14] H.-Y. Zhou, Y. Yu, C. Wang, S. Zhang, Y. Gao, J. Pan, J. Shao, G. Lu, K. Zhang, and W. Li, "A transformer-based representation-learning model with unified processing of multimodal input for clinical diagnostics," *Nature biomedical engineering*, vol. 7, no. 6, pp. 743–755, 2023.

[15] M. Löppenberg and A. Schwung, "Structured graph generation by evolutionary algorithm for program code development," in *IECON 2024 - 50th Annual Conference of the IEEE IES*, 2024, pp. 1–6.

[16] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE transactions on evolutionary computation*, vol. 6, no. 2, pp. 182–197, 2002.

[17] L. N. Asha, A. Dey, N. Yodo, and L. G. Aragon, "Optimization approaches for multiple conflicting objectives in sustainable green supply chain management," *Sustainability*, vol. 14, no. 19, p. 12790, 2022.

[18] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: Empirical results," *Evolutionary computation*, vol. 8, no. 2, pp. 173–195, 2000.

[19] D. Altinses and A. Schwung, "Performance benchmarking of multimodal data-driven approaches in industrial settings," *Machine Learning with Applications*, p. 100691, 2025.

[20] R. Das, S. Sen, and U. Maulik, "A survey on fuzzy deep neural networks," *ACM CSUR*, vol. 53, no. 3, pp. 1–25, 2020.

[21] L. Fan, "Revisit fuzzy neural network: Demystifying batch normalization and relu with generalized hamming network," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[22] S. R. Price, S. R. Price, and D. T. Anderson, "Introducing fuzzy layers for deep learning," in *2019 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. IEEE, 2019, pp. 1–6.

[23] A. D. Kulkarni, "Fuzzy convolution neural networks for tabular data classification," *arXiv preprint arXiv:2406.03506*, 2024.

[24] D. Altinses, D. O. S. Torres, A. M. Gobachew, S. Lier, and A. Schwung, "Synthetic dataset generation for optimizing multimodal drone delivery systems." *Drones (2504-446X)*, vol. 8, no. 12, 2024.