A Surrogate-Assisted Differential Evolution Using Fuzzy Logic for Expensive Constrained Optimization

Shengqi Gui*, Zusheng Tan†, Jing-Yu Ji*, and Sanyou Zeng‡
*College of Management, Shenzhen University, Shenzhen, China
shengqigui0301@gmail.com
†School of Decision Science, Lingnan University, Hong Kong
‡School of Mechanical Engineering and Electronic Information, China University of Geosciences, Wuhan, China

***Abstract*—In recent years, surrogate-assisted evolutionary algorithms have demonstrated strong capability for solving expensive constrained optimization problems. Although much of the existing literature concentrates on cases dominated by inequality constraints, expensive optimization with equality constraints remains comparatively underexplored, despite equality constraints being common in classical constrained formulations. To address this issue, this paper develops a new framework that couples a multilayer perceptron regression surrogate with a rule-based fuzzy inference mechanism and differential evolution (DE) for efficient equality-constrained search under limited evaluation budgets. Specifically, we design a type-1 Mamdani fuzzy inference system to realize a self-adaptive penalty strategy without prescribing an explicit functional form for the penalty mapping. The MLP surrogate and the fuzzy rule-based penalty function are embedded into the DE process to jointly steer the population toward feasibility. With surrogate guidance, the algorithm can allocate more search effort to exploring and refining promising feasible regions. Experimental studies further verify the effectiveness and practical potential of the proposed surrogate-assisted evolutionary approach for challenging expensive equality-constrained optimization tasks.**

***Index Terms*—fuzzy inference system, constrained optimization, multiobjective differential evolution**

I. Introduction

Expensive constrained optimization problems (ECOPs) are pervasive across many scientific and engineering applications, including aerospace and automotive design [1], [2], computational electromagnetics [3], [4], and structural engineering [5]. An ECOP typically involves a single objective function subject to one or more constraints. In these problems, both the objective and the constraints are treated as expensive black-box functions: analytical gradients are generally unavailable, and evaluating feasibility and performance requires costly simulations or physical experiments. For instance, a computational fluid dynamics run can consume minutes to hours of CPU time per evaluation. Consequently, ECOP research primarily aims to identify an optimal solution within the feasible region using a strictly limited budget of fitness evaluations (FEs). In this study, we concentrate on ECOPs characterized by inequality constraints.

For clarity, we consider an ECOP with m constraints [6]–[9], which can be expressed as

$$\begin{array}{lll} \text{minimize} & f(\mathbf{x}), & \mathbf{x}=(x_1,x_2,\ldots,x_n), \\ \text{subject to} & g_i(\mathbf{x})\leq 0, & i=1,\ldots,p, \\ & h_i(\mathbf{x})=0, & i=p+1,\ldots,p+q. \end{array} \tag{1}$$

In this formulation, each decision variable x_i is restricted to the interval $[L_i,U_i]$, where L_i and U_i denote its lower and upper bounds, respectively. The scalar function $f(\mathbf{x})$ is the objective to be minimized, and $\mathbf{x}$ is the n-dimensional decision vector. Parameters n, p, and q represent the numbers of decision variables, equalities and inequalities, respectively.

To quantify infeasibility, the aggregated constraint violation of a candidate solution $\mathbf{x}$ is defined by

$$C(\mathbf{x})=\sum_{i=1}^{m}\hat{g}_i(\mathbf{x}), \tag{2}$$

where the violation of the ith constraint is computed as

$$\hat{g}_i(\mathbf{x})=\begin{cases} \max\{g_i(\mathbf{x}),0\}, i=1,\ldots,p, \\ \max\{|h_i(\mathbf{x})-0.0001|,0\}, i=p+1,\ldots,p+q \end{cases} \tag{3}$$

Accordingly, $\mathbf{x}$ is considered feasible if and only if $C(\mathbf{x})=0$; otherwise, it is infeasible.

Evolutionary algorithms (EAs) are population-driven metaheuristics and have been extensively adopted for a wide range of optimization tasks, including constrained optimization problems (COPs). Nonetheless, their search process typically relies on a large number of FEs to achieve satisfactory convergence. As a result, when the evaluation budget is strictly limited, standard EAs often struggle to deliver solutions of acceptable

This research was supported by the Scientific Research Foundation for the Phase III Construction of a High-Level University for Youth Scholars at Shenzhen University (Grant No. 000001032933), by the Interdisciplinary Team Research Project of the College of Management, Shenzhen University (Grant No. 20240408), and by the Shenzhen Key Talent Research Start-up Grant Programme (Grant No. 8270001192). This work was done partly when Mr. Gui was with College of Management, Shenzhen University. He is now with the School of Engineering, The University of Edinburgh, Edinburgh, United Kingdom. *(Corresponding author: Jing-Yu Ji)*

quality, making their direct application to expensive optimization problems inefficient and, in some cases, infeasible. To improve efficiency under costly evaluation settings, surrogate-assisted evolutionary algorithms (SAEAs) [10]–[14] have been developed. SAEAs integrate computationally cheap surrogate models with the original objective and/or constraint functions to steer the evolutionary search. By substituting a portion of expensive real FEs with surrogate predictions, SAEAs can substantially reduce overall computational expense while maintaining effective search guidance.

In the past few years, numerous SAEAs have been proposed for computationally expensive optimization [15]–[18]. A representative line of work leverages Kriging models together with infill sampling criteria, such as expected improvement, probability of improvement, lower confidence bound, and multiobjective acquisition strategies, to identify candidate solutions for expensive evaluation. These methods exploit both the surrogate-predicted objective values and the associated predictive uncertainty estimated by Kriging. Nevertheless, the effort required to construct and update a Kriging surrogate typically grows quickly with an increasing number of training samples, which can become a computational bottleneck. Beyond Kriging-based acquisition approaches, many SAEAs also incorporate heuristic surrogate management mechanisms specifically designed to match the population-based search dynamics of evolutionary algorithms.

While significant advancements have been made in ECOPs with inequality constraints, the exploration of ECOPs with equality constraints remains relatively nascent. This gap is particularly notable given the complexities associated with approximating equality constraints. To address this, our study introduces a multilayer perceptron (MLP) surrogate-assisted DE with a Mamdani fuzzy model-based penalty function (SADE-MFZ) for solving ECOPs.

The remainder of this paper is organized as follows: Section II provides an overview of the MLP and gradient descent. Section III discusses the proposed SADE-MFZ in detail, while Section IV is dedicated to an empirical study and in-depth analysis of the proposed algorithm for ECOPs. Finally, Section V concludes the paper and outlines potential directions for future research.

II. Preliminaries

This section introduces the foundational concepts central to our study: the MLP regression model and gradient descent, along with their associated numerical methodologies.

A. Multi-layer Perceptron for Multiple-Output Regression

A MLP is a feed-forward neural network [19] that learns a nonlinear mapping from an input vector $x \in \mathbb{R}^d$ to multiple continuous targets $y \in \mathbb{R}^K$, i.e., $f : \mathbb{R}^d \to \mathbb{R}^K$. It typically consists of L fully connected hidden layers that extract a shared representation, followed by a linear output layer with K neurons. Let $h^{(0)} = x$. The ℓ-th hidden layer ($\ell = 1, \dots, L$) is computed by

$$z^{(\ell)} = W^{(\ell)} h^{(\ell-1)} + b^{(\ell)}, \qquad h^{(\ell)} = \mathrm{ReLU}\left(z^{(\ell)}\right), \tag{4}$$

where the ReLU activation is defined as $\mathrm{ReLU}(u) = \max(0, u)$ (element-wise). The regression output is produced via a linear transformation:

$$\hat{y} = W^{(L+1)} h^{(L)} + b^{(L+1)}, \qquad \hat{y} \in \mathbb{R}^K. \tag{5}$$

Thus, the general form of an L-hidden-layer MLP can be written compactly as

$$f(x;\theta) = W^{(L+1)}\, \sigma\Big(W^{(L)}\, \sigma(\cdots \sigma(W^{(1)}x + b^{(1)}) \cdots) + b^{(L)}\Big) + b^{(L+1)}, \tag{6}$$

with $\sigma(\cdot) = \mathrm{ReLU}(\cdot)$ in hidden layers and $\theta = \{W^{(\ell)}, b^{(\ell)}\}_{\ell=1}^{L+1}$. The model is commonly trained by minimizing a multi-output loss such as the (weighted) mean squared error:

$$\mathscr{L}_{\mathrm{MSE}} = \frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{K} \alpha_k \left(\hat{y}_{ik} - y_{ik}\right)^2, \tag{7}$$

where α_k can balance targets with different scales or importance.

The MLP's capability as a multi-output regression model is demonstrated by its ability to simultaneously predict multiple continuous outputs, making it well-suited for tasks requiring the prediction of various correlated outcomes from a given set of inputs. For more intricate regression tasks, MLP configurations can be enhanced with additional hidden layers and neurons, and advanced deep learning techniques may be applied for training.

B. Mamdani Fuzzy Model

Proposed by E. H. Mamdani in 1975, the Mamdani fuzzy inference system (FIS) [20] is a classic rule-based framework for representing nonlinear and poorly defined processes using linguistic knowledge. In this model, both the antecedent and consequent parts of the IF–THEN rules are described by fuzzy sets, which makes the approach particularly suitable for expert-driven decision making and control tasks. A typical Mamdani FIS proceeds through four stages: fuzzification of crisp inputs, inference via rule activation under fuzzy logic operators, aggregation of the rule-wise fuzzy outputs, and defuzzification to produce a single numerical output. The corresponding workflow is illustrated in Fig. 1.

Consider a multiple-input single-output Mamdani FIS with n inputs and one output. The ith fuzzy rule can be written as

$$\begin{aligned} R_i: \quad & \text{IF } x_1 \text{ is } A_{i1} \text{ AND } x_2 \text{ is } A_{i2} \text{ AND } \dots \text{ AND } x_n \text{ is } A_{in}, \\ & \text{THEN } y \text{ is } C_i, \end{aligned} \tag{8}$$

where A_{ij} denotes the antecedent linguistic label (fuzzy set) associated with input variable x_j, and C_i denotes the consequent linguistic label for the output variable y. The collection of rules $R = \{R_i \mid i = 1, 2, \dots, k\}$ forms the *rule base*. Together with the involved fuzzy sets $A = \{A_{ij}\}$ and $C = \{C_i\}$, the resulting description is commonly referred to as the *knowledge base*. For rule R_i, let $\mu_{ij}(x_j)$ be the membership degree of x_j in A_{ij} and $\mu_{C_i}(y)$ be the membership function of the consequent set C_i.

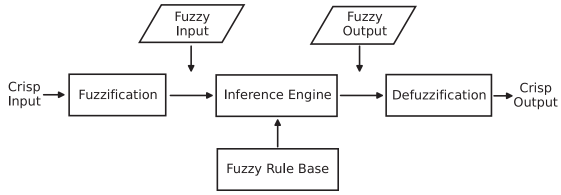

Fig. 1. Mamdani FIS.

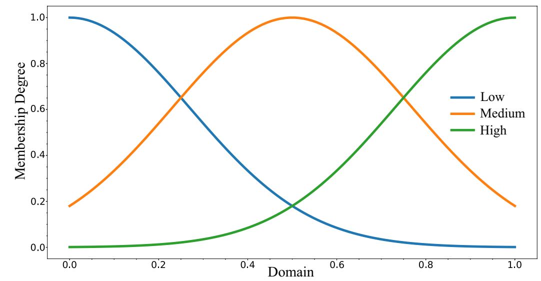

Fig. 2. Schematic diagram for the membership degree of antecedents.

Using the minimum t-norm to implement the logical conjunction, the activation (firing) level of rule R_i is computed as

$$\beta_i(y) = \mu_{i1}(x_1) \wedge \mu_{i2}(x_2) \wedge \cdots \wedge \mu_{in}(x_n) \wedge \mu_{C_i}(y), \tag{9}$$

where $\wedge$ denotes the AND operator. Equivalently, the firing level can be expressed as a function of y:

$$\beta_i(y) = \min\left(\min_{j=1,2,\ldots,n} \mu_{ij}(x_j),\, \mu_{C_i}(y) \right). \tag{10}$$

After evaluating all rules, the aggregated output fuzzy set is obtained by combining the individual rule outputs through the maximum operator, yielding the overall response

$$\mu(y) = \max_{i=1,2,\ldots,k} \beta_i(y) = \max_{i=1,2,\ldots,k} \left\{ \min\left(\min_{j=1,2,\ldots,n} \mu_{ij}(x_j),\, \mu_{C_i}(y) \right) \right\}. \tag{11}$$

To convert the aggregated fuzzy set into a crisp output, various defuzzification strategies can be adopted, such as basic defuzzification distribution, center of area, and center of gravity (COG). Under the COG criterion, the final output is computed by

$$\hat{y} = \frac{\int y\,\mu(y)\,dy}{\int \mu(y)\,dy}. \tag{12}$$

III. The Proposed Algorithm

A. FIS for Surrogate-Driven Population Evolution

Rather than introducing yet another variant of existing state-of-the-art self-adaptive penalty schemes, this study focuses on making use of representative penalty formulations to construct the rule base of our proposed FIS for surrogate-driven population evolution. The penalty term $p(\mathbf{x})$ is determined not only by the objective value $f(\mathbf{x})$ and the violation measure $v(\mathbf{x})$, but also by the proportion of feasible individuals in the current population, denoted by r_f. This motivates writing

$$p(\mathbf{x}) = g\big(v_n(\mathbf{x}), f_n(\mathbf{x}), r_f\big). \tag{13}$$

Rather than specifying an explicit analytical form for $g(\cdot)$, we approximate this mapping using a Mamdani-type fuzzy logic controller (FLC), whose inputs are given by $\big(v_n(\mathbf{x}), f_n(\mathbf{x}), r_f\big)$. Next, we define fuzzy partitions of the antecedent space and consequent space.

For the antecedent (input) domain, three linguistic terms are employed, namely *Low*, *Medium*, and *High*, whose Gaussian membership functions are parameterized by mean μ and standard deviation σ, respectively. The resulting membership functions are illustrated in Fig. 2. Similarly, the consequent (output) domain, shown in Fig. 3, is also discretized into three fuzzy sets: *LOW*, *MID*, and *HIGH*. These sets are defined using Gaussian parameters with different (μ, σ) values. Regarding the conditional relationship of $p(\mathbf{x})$ with $f_n(\mathbf{x})$, $v_n(\mathbf{x})$, and r_f, the corresponding rule base R is constructed and its detailed formulation is summarized in Table I. The output of the FLC is defuzzified using the COG method and thus, $p(x)$ is generated. The final modified objective function $F(\mathbf{x})$ can be expressed as

$$F(\mathbf{x}) = p(\mathbf{x}) + r_f * f_n(\mathbf{x}) + (1 - r_f) * v_n(\mathbf{x}). \tag{14}$$

TABLE I
Fuzzy Rule from Antecedents to Consequents.

Rules.	$f_n(\mathbf{x})$	$v_n(\mathbf{x})$	r_f	$p(\mathbf{x})$
1	Low	Any	Low	LOW
2	Low	Low	High	LOW
3	Low	Medium	High	MID
4	Medium	Low	Any	LOW
5	Medium	Medium	Low	MID
6	Medium	Medium	High	HIGH
7	Medium	High	Low	HIGH
8	High	Low	Any	MID
9	High	Medium	Low	HIGH
10	High	Medium	High	MID
11	High	High	Low	HIGH
12	Any	High	High2	HIGH

B. Overall Algorithm

The overall procedure of SADE-MFZ is summarized in **Algorithm 1**. In implementation, SADE-MFZ maintains the following data structures:

1) A population P of size NP, i.e., $P = \{\mathbf{x}_i \mid i = 1, 2, \ldots, NP\}$.
2) For each individual $\mathbf{x}_i$, its objective value and constraint-related quantities, including the per-constraint violation and the aggregated violation, denoted by $\{f(\mathbf{x}_i), \hat{g}_j(\mathbf{x}_i), C(\mathbf{x}_i) \mid i = 1, 2, \ldots, NP;\ j = 1, 2, \ldots, p + q\}$.
3) A training archive *TS* that stores all solutions evaluated by exact (expensive) FEs.

In the initialization stage, NP candidate solutions are generated within the problem-specific decision bounds using Latin hypercube sampling (LHS). Each sampled solution is then assessed by the expensive objective/constraint evaluations. The resulting decision vectors $\mathbf{x} = (x_1, x_2, \ldots, x_n)$, together with the corresponding objective and constraint information, are

Algorithm 1: SADE-MFZ

Input:
- NP: the population size;
- n: the number of variables;
- $[U,\ L]$: the upper and lower bounds of variables
- $MaxFEs$: the maximum number of expensive FEs.

Initialization:
- Initiate NP individuals (Set P) in $[U,\ L]$ by LHS;
- Evaluate the NP individuals by exact FEs;
- Set $uFE = NP$;
- Archive the evaluated NP individuals (Set TS).

Evolutionary Progress:

while $uFE < MaxFE$ **do**

————————**FIS for Surrogate-Driven:**————————

Copy the population P as P_c;

Build an MLP regression model with TS;

for *termination is not met* **do**

Use DE operators to reproduce NP offspring as Q_c;

Calculate the fuzzy rule-based penalty function for each individual in $Q_c \cup P_c$;

Use DE selection operator to update P_c via Q_c;

end

Find the best solution $\mathbf{x}_b$ from P_c;

Evaluate $\mathbf{x}_b$ via expensive FE;

Set $uFE = uFE + 1$;

Find the worst solution $\mathbf{x}_w$ from P;

if x_b *is better than* x_w **then**

Use $\mathbf{x}_b$ to update $\mathbf{x}_w$;

else

Employ the local search to the best solution from P.

end

end

Output:

The best individual $\mathbf{x}_b$ in P.

appended to TS, which subsequently serves as the feature–label dataset for training the MLP regression surrogate.

After initialization, the data-driven population evolution based on FIS is applied at each generation. In this phase, the archive containing all expensively evaluated solutions is used as the training set TS to train the MLP. The current population, consisting of NP individuals, serves as the initial population under the surrogate-driven evolution. Subsequently, the for loop operates analogously to most model-free EAs. The outputs of well-trained surrogates substitute expensive FEs to drive the population-based evolution within a certain number of generations. Specifically, for each offspring $\mathbf{u}$, the current-to-$p$best/1 mutation and binomial crossover strategies are employed. Then, the values of the objective function, constraint violations, and overall constraint violation are predicted by the surrogate. To compare $\mathbf{x}$ and $\mathbf{u}$, their modified objective function values $F(\mathbf{x})$ and $F(\mathbf{u})$ are applied. After the for loop, the evolved population P is output for further infill sampling. As DE is established techniques, its detailed application is not elaborated in **Algorithm 1**. Like the model-free algorithm, above steps are iteratively repeated till the termination is satisfied. The activation of the gradient-based local search [21] is contingent upon no improvement being observed in the MLP-based local search. The evolutionary process continues until the maximum number of expensive FEs ($MaxFEs$) is reached. Throughout this evolutionary progression, any vector that undergoes precise evaluation will have its values $\{\mathbf{x}, f(\mathbf{x}), \hat{g}_j(\mathbf{x}),\ j = 1,2\ldots m\}$ archived in the training set TS.

IV. Empirical Studies

A. Experimental Settings and Performance Metrics

In this initial exploration of ECOPs, 18 test instances with $n = 30$ from the IEEE CEC 2010 test suite [22] have been adopted. We assume the objective function and constraints of these problems to be expensive. The $MaxFEs$ is set at 1000 for each test instance. Notably, as an early effort in this field, this limitation is considered significant enough to demonstrate our capability in solving ECOPs.

The population size (PS) is fixed at 100, and the termination of for loop is set to 500. The adopted DE variant and its parameter settings are borrowed from [21], [23], [24]. Each of the 18 benchmark functions is subjected to optimization over 20 independent runs. The MLP regression model employed consists of two hidden layers, each with 50 neurons. This MLP neural network is implemented using PyTorch v1.10.0, with the AdamW optimizer chosen for minimizing the loss function. AdamW's parameters are retained at their default settings, which have been found to deliver satisfactory performance in most experiments.

To assess the performance of SADE-MFZ, we utilize three widely recognized evaluation criteria [22]: successful rate (SR), average value of constraint violation (ACV), and average value of objective function (AVF). In the context of COPs, feasible solutions are prioritized over infeasible ones. Hence, SR is employed as the primary comparative metric. When SRs are equal among the six algorithms, AVF and ACV are used for further comparison.

B. Comparison with Five State-of-the-art Algorithms

For comparative analysis, five recent SAEAs are selected: DSI-C^{2}oDE [25], MPMLS [26], SA-TSDE [27], SParEA [28], and eToSA-DE [29]. To optimize their performance, parameters for these algorithms are configured as recommended in their respective publications. Notably, DSI-C^{2}oDE, MPMLS, and SA-TSDE have been originally developed for ECOPs with only inequality constraints, making them less suitable for our testing scenario involving equality constraints. Modifying algorithms that have demonstrated success exclusively with inequality-constrained ECOPs for handling equality constraints may lead to biased results. On the other hand, few algorithms specifically designed for equality constraints exist in the literature. Consequently, these four algorithms are primarily analysed based on their performance on test instances with inequality constraints.

Table II compares SADE-MFZ with five state-of-the-art baselines in terms of success rate (SR) and average function value (AFV). Overall, SADE-MFZ exhibits the strongest robustness: it achieves SR = 1.00 on all 14 instances, whereas the compared algorithms suffer from noticeable feasibility failures on multiple problems (e.g., $C03_{30}$, $C05_{30}$, $C06_{30}$, $C09_{30}$, $C10_{30}$, $C16_{30}$, $C17_{30}$, and $C18_{30}$), often resulting in SR values far below 1.00 and even "N.A." AFV due to no feasible solution being found. This indicates that SADE-MFZ delivers consistently reliable constraint satisfaction across the

TABLE II
EMPIRICAL RESULTS OF THE FIVE COMPARED ALGORITHMS AND SADE-MFZ

Inst.	DSI-C^{2}oDE		MPMLS		SA-TSDE		SParEA		eToSA-DE		SADE-MFZ	
	SR	AFV	SR	AFV	SR	AFV	SR	AFV	SR	AFV	SR	AFV
$C01_{30}$	1.00	-3.2181E-01 +	1.00	-3.2167E-01 +	1.00	-2.7346E-01+	1.00	-2.12E-01 +	1.00	-2.0439E-01 +	1.00	-1.5095E-01
$C02_{30}$	0.00	N.A. −	0.90	4.0369E+00 −	0.90	4.1554E+00 −	0.00	N.A. −	1.00	1.9177E+00 +	1.00	4.5745E+00
$C05_{30}$	0.20	5.5638E+02 −	0.00	N.A. −	0.00	N.A. −	0.00	N.A. −	1.00	2.6014E+02 −	1.00	9.5922E+01
$C06_{30}$	0.20	3.6899E+02 −	0.00	N.A. −	0.00	N.A. −	0.00	N.A. −	1.00	2.8682E+02 −	1.00	2.4417E+02
$C07_{30}$	1.00	4.1740E+05 +	1.00	2.0297E+06 +	1.00	2.1737E+06 +	1.00	2.80E+09 −	1.00	6.46E+10 −	1.00	2.4775E+07
$C08_{30}$	1.00	4.7846E+06 +	1.00	1.4224E+08 +	1.00	1.1560E+08 +	1.00	3.50E+09 −	1.00	8.54E+10 −	1.00	7.8854E+08
$C09_{30}$	0.00	N.A. −	0.00	N.A. −	0.00	N.A. −	0.00	N.A. −	1.00	1.07E+13 −	1.00	5.2057E+10
$C10_{30}$	0.00	N.A. −	0.00	N.A. −	0.00.	N.A. −	0.00	N.A. −	1.00	1.16E+13 −	1.00	3.9268E+10
$C13_{30}$	1.00	-5.2060E+01 −	1.00	-4.5892E+01 −	1.00	-5.0718E+01−	0.84	-9.91E+00 −	0.96	-9.9175E+00 −	1.00	-5.2779E+01
$C14_{30}$	1.00	1.3251E+14 −	1.00	1.2689E+12 −	1.00	1.8844E+14 −	0.92	6.7E+13 −	1.00	4.15E+14 −	1.00	5.2854E+10
$C15_{30}$	0.90	3.0351E+14 −	0.45	1.2271E+14 −	1.00	1.1035E+15 −	0.20	1.4E+14 −	0.24	6.30E+14 −	1.00	3.6896E+12
$C16_{30}$	1.00	1.1214E+00 −	0.40	1.0447E+00 −	0.05	1.1330E+00 −	0.00	N.A. −	0.76	1.0217E+00 −	1.00	8.1296E-01
$C17_{30}$	0.25	5.8237E+02 −	0.80	6.3969E+02 −	0.00	N.A. −	0.00	N.A. −	1.00	4.4268E+01 +	1.00	5.8528E+02
$C18_{30}$	0.35	2.3748E+04 −	1.00	1.5860E+04 −	0.95	4.0749E+04 −	0.00	N.A. −	1.00	4.5972E+01 −	1.00	2.5001E+01
Cmp.	**Wins: 3; Lose: 11**		**Wins: 3; Lose: 11**		**Wins: 3; Lose: 11**		**Wins: 1; Lose: 13**		**Wins: 3; Lose: 11**		NaN	

whole test suite, while DSI-C^{2}oDE, MPMLS, and SA-TSDE show similar limitations in robustness and SParEA is the least reliable (frequent SR < 1.00).

When all methods succeed (i.e., SR ties at 1.00), AFV highlights solution quality differences. SADE-MFZ achieves clear AFV advantages on several challenging instances, notably $C14_{30}$ and $C15_{30}$, where it attains much smaller objective values than all competitors, and it also improves over SParEA on $C07_{30}$ and $C08_{30}$ and over eToSA-DE on many high-magnitude cases (e.g., $C07_{30}$ to $C10_{30}$ and $C14_{30}$). However, SADE-MFZ is not uniformly best in AFV under tied SR: DSI-C^{2}oDE/MPMLS/SA-TSDE report smaller AFV on $C01_{30}$ and also on $C07_{30}$ and $C08_{30}$, suggesting that these methods can be competitive in objective minimization once feasibility is guaranteed. In summary, the empirical results demonstrate that SADE-MFZ provides the best overall trade-off, near-perfect reliability (SR) with strong AFV performance on hard instances, which explains its favorable win-loss outcomes against the five baselines.

Based on these comparative analyses, it can be concluded that SADE-MFZ is highly competitive with state-of-the-art SAEAs specifically designed for ECOPs, and notably more applicable for effectively solving ECOPs encompassing equality constraints.

C. *Effectiveness of FIS*

The FIS has been widely demonstrated to be effective in handling uncertainty. However, very few studies have leveraged the advantages of FIS to address the uncertainty of surrogate model in expensive constrained optimization. To investigate the influence of our proposed FIS, we discard FIS, resulting in a variant called SADE-WO-MFZ. For a straightforward comparison, we use the CEC2010 test set due to its more difficult and complex problem characteristics. Table III summarizes the significant differences in SR and AFV values obtained from SADE-MFZ and SADE-WO-MFZ on 14 test instances.

TABLE III
EMPIRICAL RESULTS OF SADE-WO-MFZ AND SADE-MFZ

Inst.	SADE-WO-MFZ		SADE-MFZ	
	SR	AFV	SR	AFV
$C01_{30}$	1.00	-1.3083E-01 −	1.00	-1.5095E-01
$C02_{30}$	1.00	4.1918E+00 +	1.00	4.5745E+00
$C05_{30}$	1.00	3.6671E+02 −	1.00	9.5922E+01
$C06_{30}$	1.00	4.6200E+02 −	1.00	2.4417E+02
$C07_{30}$	1.00	7.0983E+06 +	1.00	2.4775E+07
$C08_{30}$	1.00	8.9143E+06 +	1.00	7.8854E+08
$C09_{30}$	1.00	6.1935E+12 −	1.00	5.2057E+10
$C10_{30}$	1.00	6.7285E+12 −	1.00	3.9268E+10
$C13_{30}$	1.00	-3.5459E+01 −	1.00	-5.2779E+01
$C14_{30}$	1.00	4.9412E+10 +	1.00	5.2854E+10
$C15_{30}$	1.00	3.1215E+14 −	1.00	3.6896E+12
$C16_{30}$	1.00	9.1545E-01 −	1.00	8.1296E-01
$C17_{30}$	0.80	3.8921E+02 −	1.00	5.8528E+02
$C18_{30}$	1.00	2.6798E+01 −	1.00	2.5001E+01
Cmp.	**Wins: 4; Lose: 10**		NaN	

According to Table III, both variants are highly reliable in terms of feasibility, as they achieve SR $= 1.00$ on 13 out of 14 instances. The only exception is $C17_{30}$, where removing the type-1 Mamdani fuzzy inference system causes a clear robustness drop (SADE-WO-MFZ: SR $= 0.80$ vs. SADE-MFZ: SR $= 1.00$). This indicates that the Mamdani-based mechanism contributes primarily to stabilizing the search and maintaining consistent feasibility, especially on instances where the optimization process is more prone to premature failure.

In terms of solution quality (AFV, smaller is better), SADE-MFZ outperforms SADE-WO-MFZ on 9 out of 14 instances, notably showing substantial improvements on several difficult cases such as $C05_{30}$, $C06_{30}$, $C09_{30}$, $C10_{30}$, and $C15_{30}$, where the AFV reductions are large (often by orders of magnitude). Meanwhile, SADE-WO-MFZ is better on 5 instances ($C02_{30}$, $C07_{30}$, $C08_{30}$, $C14_{30}$, and $C17_{30}$), suggesting that discarding the fuzzy inference can sometimes yield more competitive objective values on a few problems, but at the cost of weaker overall consistency. Overall, the comparison ("Wins: 4; Lose:

10” for SADE-WO-MFZ) supports that incorporating the type-1 Mamdani fuzzy inference system generally delivers a more robust and better-quality optimizer across the benchmark set.

V. Conclusion

This study proposes SADE-MFZ, a surrogate-assisted evolutionary algorithm tailored for ECOPs. By integrating an MLP surrogate model and a type-1 Mamdani FIS into the DE framework, SADE-MFZ jointly enhances constraint handling and objective optimization. A set of representative 30-dimensional ECOP test instances was employed to assess its effectiveness. The empirical results indicate that SADE-MFZ consistently attains feasible solutions with competitive objective values, demonstrating strong robustness and solution quality. In comparison with several state-of-the-art SAEAs, SADE-MFZ achieves overall superior performance, with particularly notable gains in constraint satisfaction. Moreover, ablation studies using SADE-MFZ variants (e.g., removing the type-1 Mamdani FIS) further confirm the contribution of the proposed fuzzy mechanism to the algorithm’s effectiveness.

Future work will investigate type-2 fuzzy inference systems within surrogate-assisted evolutionary optimization to better accommodate higher uncertainty and more complex constraint landscapes, thereby extending the applicability and capability of SADE-MFZ on more challenging ECOPs.

References

[1] X. Peng, X. Li, X. Chen, X. Chen, and W. Yao, “A hybrid deep learning framework for unsteady periodic flow field reconstruction based on frequency and residual learning,” *Aerospace Science and Technology*, vol. 141, p. 108539, 2023.

[2] J.-Y. Ji, Z. Tan, M.-L. Wong, and J. Zhang, “Heat-pipe-constrained iot device layout via multiobjective differential evolution,” *IEEE Internet of Things Journal*, vol. 12, no. 7, pp. 8261–8275, 2025.

[3] Y. Zhang, X.-F. Ji, X.-Z. Gao, D.-W. Gong, and X.-Y. Sun, “Objective-constraint mutual-guided surrogate-based particle swarm optimization for expensive constrained multimodal problems,” *IEEE Transactions on Evolutionary Computation*, vol. 27, no. 4, pp. 908–922, 2023.

[4] H. Zou, S. Zeng, C. Li, and J. Ji, “A survey of machine learning and evolutionary computation for antenna modeling and optimization: Methods and challenges,” *Engineering Applications of Artificial Intelligence*, vol. 138, p. 109381, 2024.

[5] J. Hong, Z.-H. Zhan, L. He, Z. Xu, and J. Zhang, “Protein structure prediction using a new optimization-based evolutionary and explainable artificial intelligence approach,” *IEEE Transactions on Evolutionary Computation*, vol. 29, no. 3, pp. 646–660, 2025.

[6] R. Mallipeddi and P. N. Suganthan, “Ensemble of constraint handling techniques,” *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 4, pp. 561–579, 2010.

[7] J.-Y. Ji, S. Zeng, and M. L. Wong, “ε-constrained multiobjective differential evolution using linear population size expansion,” *Information Sciences*, vol. 609, pp. 445–464, 2022.

[8] C. A. Coello Coello, “Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art,” *Computer Methods in Applied Mechanics and Engineering*, vol. 191, no. 11, pp. 1245–1287, 2002.

[9] J. Zhang, J.-Y. Ji, and S. Kwong, “A surrogate-assisted differential evolution for expensive optimization with equality constraints,” *Applied Soft Computing*, vol. 189, p. 114547, 2026.

[10] K. Yu, F. Chen, J. Liang, M. Yu, K. Chen, C. Yue, and Y. Bi, “Objective-constraint correlation-guided evolutionary direction adaptive adjustment for expensive constrained optimization,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 55, no. 8, pp. 5758–5772, 2025.

[11] J.-Y. Ji, Z. Tan, S. Zeng, E. W. K. See-To, and M.-L. Wong, “A surrogate-assisted evolutionary algorithm for seeking multiple solutions of expensive multimodal optimization problems,” *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 8, no. 1, pp. 377–388, 2024.

[12] Y. Su, L. Xu, and E. D. Goodman, “Hybrid surrogate-based constrained optimization with a new constraint-handling method,” *IEEE Transactions on Cybernetics*, vol. 52, no. 6, pp. 5394–5407, 2022.

[13] Y. Liu, J. Liu, and L. Tang, “Expensive constrained optimization via adaptive switching between surrogate-assisted constrained and unconstrained search,” *IEEE Transactions on Evolutionary Computation*, pp. 1–1, 2025.

[14] J.-Y. Ji, B. Chiu, Z. Tan, B. Niu, L. Liang, and S. Kwong, “Surrogate-assisted evolutionary algorithm for expensive optimization with equality and inequality constraints,” *IEEE Transactions on Evolutionary Computation*, pp. 1–1, 2025.

[15] Z. Wang, L. Xie, G. Li, W. Gao, M. Gong, and L. Wang, “Customized evolutionary expensive optimization: Efficient search and surrogate strategies for continuous and categorical variables,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 55, no. 3, pp. 2196–2210, 2025.

[16] S. Gui, H. Liu, J.-Y. Ji, and J. Zhang, “Multi-output neural network-assisted differential evolution for optimization problems with expensive objective function and inequalities,” in *2025 12th International Conference on Machine Intelligence Theory and Applications (MiTA)*, 2025, pp. 1–6.

[17] H. Wu, Q. Chen, C. Yang, J. Ding, and Y. Jin, “A surrogate-assisted coevolutionary algorithm for expensive constrained multiobjective optimization,” in *2024 IEEE Congress on Evolutionary Computation (CEC)*, 2024, pp. 1–8.

[18] Y. Wang, D.-Q. Yin, S. Yang, and G. Sun, “Global and local surrogate-assisted differential evolution for expensive constrained optimization problems with inequality constraints,” *IEEE Transactions on Cybernetics*, vol. 49, no. 5, pp. 1642–1656, 2019.

[19] C. M. Bishop, *Neural networks for pattern recognition*. Oxford university press, 1995.

[20] E. H. Mamdani and S. Assilian, “An experiment in linguistic synthesis with a fuzzy logic controller,” *International Journal of Man-Machine Studies*, vol. 7, no. 1, pp. 1–13, 1975.

[21] T. Takahama and S. Sakai, “Constrained optimization by the ε constrained differential evolution with an archive and gradient-based mutation,” in *IEEE Congress on Evolutionary Computation*. IEEE, 2010, pp. 1–9.

[22] J. J. Liang, T. P. Runarsson, E. Mezura-Montes, M. Clerc, P. N. Suganthan, C. C. Coello, and K. Deb, “Problem definitions and evaluation criteria for the cec 2006 special session on constrained real-parameter optimization,” *Journal of Applied Mechanics*, vol. 41, no. 8, pp. 8–31, 2006.

[23] J.-Y. Ji, Z. Tan, S. Zeng, and M.-L. Wong, “An ε-constrained multiobjective differential evolution with adaptive gradient-based repair method for real-world constrained optimization problems,” *Applied Soft Computing*, vol. 152, p. 111202, 2024.

[24] J. Zhang, B. Chiu, J.-Y. Ji, L. Liang, J. Zhang, and S. Kwong, “Region division and merging-based multiobjective optimization for multimodal problems,” *IEEE Transactions on Consumer Electronics*, vol. 71, no. 3, pp. 7651–7661, 2025.

[25] Z. Song, H. Wang, and Y. Jin, “A surrogate-assisted evolutionary framework with regions of interests-based data selection for expensive constrained optimization,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 53, no. 10, pp. 6268–6280, 2023.

[26] G. Li and Q. Zhang, “Multiple penalties and multiple local surrogates for expensive constrained optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 25, no. 4, pp. 769–778, 2021.

[27] Y. Liu, J. Liu, Y. Jin, F. Li, and T. Zheng, “A surrogate-assisted two-stage differential evolution for expensive constrained optimization,” *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 7, no. 3, pp. 715–730, 2023.

[28] K. H. Rahi, H. K. Singh, and T. Ray, “Partial evaluation strategies for expensive evolutionary constrained optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 25, no. 6, pp. 1103–1117, 2021.

[29] F.-F. Wei, W.-N. Chen, W. Mao, X.-M. Hu, and J. Zhang, “An efficient two-stage surrogate-assisted differential evolution for expensive inequality constrained optimization,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 1–14, 2023.