International Journal of Engineering and Information Systems (IJEAIS)
ISSN: 2643-640X
Vol. 9 Issue 9 September - 2025, Pages: 5-16

A MultObjective Grey Wolf Optimizer Based on Fuzzy Optimality

Ahmed Abdulhusseim Jabbar

(ahmed.jabbar.pure246@student/uobabylon.edu.iq)

Ministry of Education General Directorate of Education in Anjaf Iraq

Abstract: *This paper explores MultObjective optimization (MOO), a critical methodology for identifying optimal solutions across various domains such as economics, finance, and engineering. We highlight the significance of MOO in enhancing bioeconomic models for fisheries and its application in financial analysis through the niched-Pareto genetic algorithm. Furthermore, the research examines political dynamics via centrality measures in network models and optimizes mechanical engineering designs, specifically shell and tube heat exchangers. Various MOO methods, including the global criterion method, weighted-sum method, ε-constraint method, lexicographic method, and MultObjective evolutionary algorithms (MOEAs) are discussed. Additionally, we present Fuzzy Logic as a framework for managing uncertainty and imprecision, emphasizing its core components, such as membership functions and fuzzy rules. Lastly, we introduce the Grey Wolf Optimization (GWO) algorithm inspired by natural systems, addressing its development and applications. This research aims to create a MultObjective Grey Wolf Optimizer that integrates Fuzzy Optimality, enhancing the effectiveness of solving complex MultObjective optimization challenges.*

Introduction

Optimization processes aim to identify the best solution by maximizing or minimizing specific values. When multiple objectives are involved, the problem is categorized as MultObjective optimization (MOO) [1]. This approach is widely utilized across various domains, including mathematics, engineering, social sciences, economics, agriculture, aviation, and automotive industries [2].

In the field of economics, MOO enhances bioeconomic models for fisheries, as illustrated by Mardle, Pascoe, and Tamiz [3]. Their research focuses on optimizing resource use and evaluating management effectiveness in North Sea fisheries, emphasizing four main objectives: maximizing profits, preserving historical quota distributions among countries, maintaining industry employment, and minimizing waste.

In finance, the niched-Pareto genetic algorithm (NPGA) is used to identify significant patterns within financial time series [4]. This algorithm balances two objectives—match quality and area—helping determine whether market trends indicate upward or downward movements, or patterns like head-and-shoulders.

Political applications of MOO include identifying key players within campaign dynamics, as explored by Gunasekara, Mehrotra, and Mohan [5]. Their study examines centrality measures and distances within network models, such as the Dolphin and Prisoners networks, to optimize the selection of influential actors.

In mechanical engineering, research has addressed MultObjective optimization for shell and tube heat exchangers [6]. The goal is to minimize total costs, including capital investment and operational expenditures, while also reducing the heat exchanger's length through genetic algorithm (GA) techniques.

Numerous methods exist for tackling MOO challenges. The global criterion method transforms multiple objectives into a single optimization problem by minimizing the distance between various reference points [7]. Meanwhile, the weighted-sum method aggregates multiple objectives using a normalized weight vector, though it faces issues like weight selection bias and nonconvexity [8].

To address complexities in nonconvex situations, the ε-constraint method optimizes one objective while imposing constraints on others [9]. By adjusting the ε vector, multiple optimal solutions can be explored, although certain vectors may not yield feasible results.

The lexicographic method prioritizes objectives according to decision-makers' preferences, optimizing each in order of importance until a satisfactory solution is achieved [10].

Finally, MultObjective evolutionary algorithms (MOEAs) are stochastic methods designed to find optimal Pareto solutions [11]. Unlike traditional evolutionary algorithms, some MOEAs utilize dominance relationships to guide the selection of superior solutions across generations.

This research aims to develop a novel software tool in Python for facilitating MOO applications, enhancing capabilities for solving complex optimization problems across various fields.

Fuzzy Logic serves as a valuable framework for tackling situations marked by imprecision and uncertainty [1]. Unlike conventional binary logic, which limits truth values to either true or false, Fuzzy Logic accommodates a spectrum of truth values ranging from 0 to 1 [2]. This multi-valued approach is essential for navigating ambiguity and uncertainty in decision-making.

By allowing for partial truths, Fuzzy Logic mirrors the complexities of real-life situations, facilitating a continuum of truth values rather than strict true or false classifications [3]. Its applications are diverse, encompassing areas such as control systems, image processing, natural language processing, medical diagnosis, and artificial intelligence [4].

Central to Fuzzy Logic is the membership function, which measures the extent to which an input belongs to a particular set [1]. This function assigns degrees of membership between 0 and 1, where 0 signifies non-membership and 1 indicates full membership. Fuzzy Logic provides a mathematical foundation for handling vagueness and uncertainty, enabling the representation of partial truths with wide-ranging applications. Its principles are grounded in the membership function and are executed through Fuzzy Rules [5].

Throughout history, humanity has sought to improve life by uncovering the secrets of nature. Recently, concepts from nature, such as the social structures of ant colonies, packs of grey wolves, and elephants, have inspired optimization algorithms [2]. One such technique, the Grey Wolf Optimization (GWO) algorithm, is based on the socio-hierarchical behavior of the Canis Lupus (Grey Wolf). This paper presents a comprehensive review of the GWO algorithm, detailing its development and diverse applications in addressing complex real-world problems.

This research focuses on creating a MultObjective Grey Wolf Optimizer that integrates Fuzzy Optimality, aiming to enhance the effectiveness of solving MultObjective optimization challenges.

Definitions and Properties

Definition 1: Pareto Dominance

Let ($x = (x_1, x_2, \dots, x_k)$) and ($y = (y_1, y_2, \dots, y_k)$) be two vectors. Vector (x) is said to dominate vector (y) (denoted as ($x \nprec y$)) if for all ($i \in \{1, 2, \dots, k\}$), the following conditions hold: ($f(x_i) \geq f(y_i)$) and there exists at least one (i) such that ($f(x_i) > f(y_i)$) [1].

Definition 2: Pareto_Optimality

A solution ($\boldsymbol{x} \in \boldsymbol{X}$) is considered Pareto_optimal if for every ($\boldsymbol{y} \in \boldsymbol{X}$), it is true that ($F(y) \nprec F(x)$) [2].

Definition 3: Pareto_Optimal Set

The collection of all nondominated solutions is referred to as the Pareto_optimal set, denoted as:

[$\boldsymbol{P_s} = \{ \boldsymbol{x}, \boldsymbol{y} \in \boldsymbol{X} \mid \exists \boldsymbol{F}(\boldsymbol{y}) \prec \boldsymbol{F}(\boldsymbol{x}) \}$] [**3**].

Definition 4: Pareto Optimal Front

The set that includes the objective values of the Pareto_optimal solutions is known as the Pareto_optimal front, defined as:

$P_f = \{ F(x) \mid x \in P_s \}$ [4].

You can replace the placeholders in the references with the actual citations as needed.

MultObjective Grey Wolf Optimizer (MOGWO)

The Grey Wolf Optimizer (GWO) algorithm, developed by Mirjalili, Mirjalili, and Lewis in 2014, is inspired by the social hierarchy and hunting behaviors of grey wolves. Within this framework, the optimal solution is referred to as the alpha (α) wolf, while the second and third most effective solutions are identified as beta (β) and delta (δ) wolves, respectively. The other solutions are categorized as omega (ω) wolves. The optimization process is directed by the α, β, and δ wolves, with the ω wolves trailing these leaders in their quest for the global optimum.

The hunting behavior of grey wolves is modeled through the following equations:

$$D = C \cdot \left(X_{p(t)} - X(t) \right)$$
$$X(t + 1) = X_p(t) - A \cdot D$$

Based on the context provided, it seems you are asking me, Claude, the AI assistant created by Anthropic, to paraphrase and update the information related to the variables T, A, C, Xp, and X in the given expression.

Where:

T represents the current iteration or time step

A and C are coefficient or scaling vectors that control the movement and behavior of the grey wolf

Xp is the position vector of the prey or target

X is the position vector of the grey wolf

The variables in this expression likely relate to some kind of optimization or search algorithm, such as the Grey Wolf Optimizer (GWO) algorithm, which mimics the social hierarchy and hunting behavior of grey wolves. The goal is typically to find the optimal solution or position (represented by X) based on the positions of the prey (Xp) and the coefficients A and C that guide the movement of the grey wolves.

The coefficient vectors A and C are defined as follows:

$$A = 2a \cdot (r1 - a)$$
$$C = 2 \cdot r2$$

Where:

- The elements of a*a* decrease linearly from 2 to 0 over the iterations
- r1 and r2 are random vectors within the interval [0, 1]

The GWO algorithm utilizes this simulated social structure and encircling behavior to identify optimal solutions for various optimization challenges. It keeps track of the three best solutions found thus far, encouraging the other search agents (the ω wolves) to adjust their positions in relation to these leaders. The equations are continuously applied to each search agent throughout the optimization process, simulating the hunting dynamics and exploring promising areas within the search space.

MultObjective Optimization using Grey Wolf Optimization

The Grey Wolf Optimizer (GWO) algorithm, introduced by Mirjalili, Mirjalili, and Lewis in 2014, is inspired by the social structure and hunting strategies of grey wolves. Within this model, the most effective solution is represented by the alpha (α) wolf, while the second and third best solutions are referred to as beta (β) and delta (δ) wolves, respectively. The remaining candidate solutions are categorized as omega (ω) wolves. The optimization process is directed by the α, β, and δ wolves, with the ω wolves following these leaders in their quest for the global optimum.

Exploration:
The coefficient vector A helps facilitate the exploration process, The process involves generating random values that are either greater than 1 or less than -1, which encourages the search agents to move away from the prey. Furthermore, the C vector produces random values within the interval [0, 2], facilitating stochastic emphasis (when C > 1) or de-emphasis (when C < 1) of the prey's impact on the distance computations in Equation (3.1). This enhances the algorithm's exploratory behavior, reducing the likelihood of getting stuck in local optima. Unlike A, the C parameter does not decrease linearly, ensuring random values throughout the optimization process.

Exploitation:
The exploitation phase of the GWO algorithm occurs when |A| < 1. In this case, the random values of A fall within [-1, 1], allowing search agents to position themselves anywhere between their current location and the prey's position, thereby converging towards the estimated location defined by the α, β, and δ solutions.

This balance between exploration and exploitation is a key feature of the GWO algorithm, enabling it to effectively search the solution space and converge towards the global optimum.

Optimization Process

The optimization process starts by generating an initial population of random solutions. Throughout the procedure, the top three solutions are retained as the α, β, and δ leaders. For each ω wolf, the position update formulas are applied, while the parameters (a) and (A) decrease linearly over the iterations. This adjustment helps the search agents to diverge when ($|A| > 1$) and converge when ($|A| < 1$). Once the termination criteria are satisfied. At the end of the optimization process, the position and score of the alpha (α) solution are reported as the best results.

Supplementary Formulas

$$-\left(\boldsymbol{D_\alpha} = \boldsymbol{C_1} \cdot (\boldsymbol{X_\alpha} - \boldsymbol{X}) \right)$$

$$-\left(\boldsymbol{D_\beta} = \boldsymbol{C_2} \cdot (\boldsymbol{X_\beta} - \boldsymbol{X}) \right)$$

$$-\left(\boldsymbol{D_\delta} = \boldsymbol{C_3} \cdot (\boldsymbol{X_\delta} - \boldsymbol{X}) \right)$$

$$-\left(\boldsymbol{X_1} = \boldsymbol{X_\alpha} - \boldsymbol{A_1} \cdot \boldsymbol{D_\alpha} \right)$$

$$-\left(\boldsymbol{X_2} = \boldsymbol{X_\beta} - \boldsymbol{A_2} \cdot \boldsymbol{D_\beta} \right)$$

$$-\left(\boldsymbol{X_3} = \boldsymbol{X_\delta} - \boldsymbol{A_3} \cdot \boldsymbol{D_\delta} \right)$$

$$-\left(\boldsymbol{X(t+1)} = \boldsymbol{X_1} + \boldsymbol{X_2} + \boldsymbol{X_3} \right)$$

In order to enhance Multi-Objective optimization utilizing Grey Wolf Optimization (GWO), two additional components are introduced, reflecting those present in Multi-Objective Particle Swarm Optimization (MOPSO) **[1]:**1. An archive that stores nondominated Pareto optimal solutions encountered so far.

2. A leader selection strategy that identifies the α, β, and δ solutions from the archive for the optimization process.

These components enhance the GWO algorithm's ability to explore the objective space efficiently and converge toward the Pareto optimal front in MultObjective contexts.

The archive acts as a repository for nondominated Pareto optimal solutions, managed by an archive controller that regulates entries and exits. It has a maximum capacity, and during each iteration, new nondominated solutions are compared to those already stored. Three scenarios may occur:

1. The new solution is dominated by at least one existing member; it is discarded.

2. The new solution dominates one or more members; those dominated are removed, and the new solution is accepted.

3. If neither dominates, the new solution is added to the archive.

When the archive reaches its capacity, a grid mechanism is employed to reorganize the objective space. This mechanism identifies the most crowded segment and removes a solution from it, allowing the new solution to be inserted into the least populated segment, This enhances the diversity of the estimated Pareto optimal front. The likelihood of eliminating a solution is directly linked to the number of solutions present in the corresponding hypercube (segment). If a new solution is located outside the current hypercubes, the grid mechanism expands all segments to accommodate it, which may also shift the positions of other solutions as the objective space is updated.

The grid mechanism aims to maintain a balanced distribution of solutions within the objective space, ensuring that the archived solutions provide a diverse and well-distributed approximation of the Pareto optimal front.

MultObjective Grey Wolf Optimization (MOGWO) Method

Initialization:

Set the grey wolf population X_i, where i = 1, 2, ..., n.

Initialize parameters a, A, and C.

Compute the objective values for each search agent.

Initialize Archive:

Identify nondominated solutions and initialize the archive with them.

Select Leaders:

X_α = SelectLeader(archive)

X_β = SelectLeader(archive)

X_δ = SelectLeader(archive)

Add X_α and X_β back to the archive.

Main Loop:

Set iteration counter t = 1.

While t < Max iterations:

a. Update Positions:

For each search agent, update the position using equations (3.5) to (3.11).

b. Update Parameters:

Update a, A, and C.

c. Calculate Objective Values:

Compute the objective values for all search agents.

d. Update Archive:

Identify nondominated solutions.

Update the archive based on new nondominated solutions.

e. Check Archive Capacity:

If the archive is full:

i. Execute the grid mechanism to remove one member from the archive.

ii. Add the new solution to the archive.

f. Hypercube Updates:

If any new solutions lie outside the hypercubes:

i. Update the grids to include the new solution(s).

g. Select Leaders Again:

X_α = SelectLeader(archive)

X_β = SelectLeader(archive)

X_δ = SelectLeader(archive)

Add X_α and X_β back to the archive.

h. Increment t by 1.

Return the final archive.

Output the archive containing the Pareto optimal solutions.

Equations are not provided in the text, but they refer to the position update formulas for the search agents in the MOGWO algorithm.

4. Experimental Configuration

The MultObjective Grey Wolf Optimizer (MOGWO) algorithm is evaluated in comparison to two well-established algorithms: MultObjective Particle Swarm Optimization (MOPSO) and MultObjective Evolutionary Algorithm based on Decomposition (MOEA/D). The initial parameters established for MOPSO are as follows:

The parameter settings were as follows:

- $\phi_1 = \phi_2 = 2.05$
- $\phi = \phi_1 + \phi_2$
- $w = \chi = 2$ (inertia weight, calculated as $w = \phi - 2 + \phi_2 - 4\phi$)
- $c_1 = \chi \cdot \phi_1$
- $c_2 = \chi \cdot \phi_2$
- alpha = 0.1 (grid expansion parameter)
- beta = 4 (leader selection intensity parameter)
- n_{Grid} = 10 (number of grids per axis)

For the MOEA/D the algorithm employed the following parameters:

- N = 100 (number of sub-tasks)
- T = 0.1N (number of adjacent entities)
- n_r = 0.01N (maximum instances of a new offspring during updates)
- delta = 0.9 (likelihood of choosing parents from the neighborhood)
- C_R = F = 0.5 (mutation frequencies)
- eta = 30 (distribution parameter)

All experiments involved the use of one hundred search agents and were conducted for a maximum of three thousand iterations. The research focused on ten standard multi-objective test problems selected from the CEC 2009 benchmark suite. These benchmark problems are recognized for their complexity and encompass a range of multi-objective search spaces, featuring various characteristics of Pareto optimal fronts, including convex, nonconvex, discontinuous, and multi-modal traits. For the purpose of performance evaluation, the researchers applied the Inverted Generational Distance (IGD) metric to assess convergence. Furthermore, they utilized Spacing (SP) and Maximum Spread (MS) to evaluate coverage. The mathematical formulation of IGD closely resembles that of the Generational Distance (GD), thereby providing a reliable measure for assessing the performance of the algorithms.

UF1:

$$\mathrm{f1} = \mathrm{x1} + \left(\frac{2}{|\,Ji\,|}\right) \sum_{\mathrm{j \in Ji}} \left(xj - sin\left(6\pi x1 + \frac{i\pi}{n}\right)\right)^2$$

$$j1 = \{j \mid j \; is \; odd, 2 \le j \le n\}$$
$$J2 = \{j \mid j \; is \; even, 2 \le j \le n\}$$

UF2:

$$f_1 = x_1 + (\frac{2}{|\,J_1\,|}) \sum_{j \in J1} y\, j_2$$

$$f_2 = 1 - \sqrt{x} + (\frac{2}{|\,f_2\,|}) \sum_{j \in J_2} yj_2$$

$$j_1 = \{j \mid j \; is \; odd, 2 \le j \le n\}$$

$$J_2 = \{j \mid j \; is \; even, 2 \le j \le n\}$$

$$y_j = x_j - (0.3\, x12\, cos(24\pi x_1 + \frac{4j\pi}{n}) + 0.6x_1)\, cos(6\pi x_1 + \frac{(j\pi)}{n}) \; if \; j \in J_1$$

UF3:

$$f_1 = x_1 + (\frac{2}{|\,y_1\,|})\,(4 \sum_{j \in J_1} y_{j_2} - 2 \prod_{j < j_1} \cos(20 y_j \frac{\pi}{\sqrt{j}}) + 2)$$

$$f_2 = \sqrt{x}_1 + (\frac{2}{|\,y_2\,|})\,(4 \sum_{j < j_2} y_{j_2} - 2 \prod_{j < j_2} cos(20 y_j \frac{\pi}{\sqrt{j}}) + 2)$$

$$y_j = x_j - x_1(0.5(1,0 + \frac{l(j-2)}{(n-2)}))$$

$$j = 2, 3, \ldots, n$$

UF4 :

$$f_1 = x_1 + 2 \sum_{j < j_1} h(y_j)$$

$$f_2 = 1 - x_2 + 2 \sum_{j \in I_2} h(y_j)$$

UF5:

J1, J2 are the same as in UF1

$$y_j = x_j - \sin\left(6\pi x1 + \frac{j\pi}{n}\right)$$

j = 2, 3, ..., n

$$f_1 = x_1 + ((\frac{1}{2N}) + \epsilon)\,|\,sin(2N\pi x_1)\,| + \left(\frac{2}{|\,fi\,|}\right) \sum_{j \in f_i} h(y_i)$$

$$h(t) = \frac{|\ t\ |}{(1\ +\ e_2\ |\ t\ |)}$$

$$f_1\ =\ 1\ -\ x_1\ +\ (\left(\frac{1}{2N}\right)\ +\ \epsilon)\ |\ sin(2N\pi x_1)\ |\ +2\ \sum_{j\in I\sim 2} h(yi)$$

UF6:

$$F_2\ =\ 1\ -\ x_1\ +\ max\{0, 2(\left(\frac{1}{2N}\right)\ +\ \epsilon)\ sin(2N\pi x_1)\}\ \cdot\ (\frac{2}{|\ y_2\ |})\ (4\ \sum_{j<j_2} y_{j_2}\ -\ 2\ \prod_{j<j_2}\ cos(20y_j\frac{\pi}{\sqrt{j}})\ +\ 1)$$

$$y_j\ =\ x_j\ -\ sin(6\pi x_1\ +\ \frac{(j\pi)}{n})$$

UF7:

$$f_1\ =\ (x_1)1/3\ +\ \left(\frac{2}{|\ yi\ |}\right)\sum_{j\in Ji} y_{j2}$$

$$f_2\ =\ 1\ -\ (x_1)1/3\ +\ max\{0, 2((\frac{1}{2}N)\ +\ \epsilon)\ sin(2N\pi x_1)\}\ \cdot\ \left(\frac{2}{|\ y2\ |}\right)\left(4\ \sum_{j<j2} yj_2\ -\ 2\ \prod_{j<j2} \cos\left(20\ y_j\frac{\pi}{\sqrt{j}}\right) +\ 1\right), y_j\ =\ x_j\ -\ sin(6\pi x_1\ + \frac{i\pi}{n})$$

$$j\ =\ 2, 3, \ldots, n$$

Tri-objective Test Problems:

UF8:

$$f_1\ =\ cos(0.5\ x_1\ \frac{\pi}{n})\ cos(0.5\ x_2\ \frac{\pi}{n})\ +\ 2\ -\ \left(\frac{2}{|\ J1\ |}\right)\sum_{j\in J1}(x_j\ -\ 2x_2 sin(2\pi x1\ +\frac{j\pi}{n}))2$$

$$f_2 = \cos\left(\frac{0.5x_1\pi}{n}\right)\sin\left(\frac{0.5x_2\pi}{n}\right) +\ 2\ -\ \left(\frac{2}{|\ J2\ |}\right)\ \sum_{j\in J2}\left(xj\ - 2x_2\sin\left(2\pi x_1\ +\frac{j\pi}{n}\right)\right)2$$

$$f_3\ =\ sin(0.5x_1\pi/n)\ +\ 2\ -\ (\frac{2}{|\ J3\ |})\ \sum_{j\in J3}(xj\ -\ 2x_2 sin(2\pi x_1\ +\frac{j\pi}{n}))2$$

$$J_1\ =\ \{j\mid 3\ \le\ j\ \le\ n, and\ j\ -\ 1\ is\ a\ multiple\ of\ 3\}$$

$$J_2\ =\ \{j\mid 3\ \le\ j\ \le\ n, and\ j\ -\ 2\ is\ a\ multiple\ of\ 3\}$$

$$J_3\ =\ \{j\mid 3\ \le\ j\ \le\ n, and\ j\ is\ a\ multiple\ of\ 3\}$$

UF9:

$$f_1\ =\ 0.5[(max\{0, (1\ +\ 0.1)(1\ -\frac{4(2x_1\ -\ 1)}{n2})\})\ +\ 2x_1]x2\ +\ 2\ -\ (\frac{2}{|\ J1\ |})\ \sum_{j\in J1}(xj\ -\ 2x_2 sin(2\pi x_1\ +\frac{j\pi}{n}))2$$

$$f2 = 0.5\left[\left(max\left\{0,(1 + 0.1)\left(1 - \frac{4(2x1 - 1)}{n2}\right)\right\}\right) + 2x1\right]x_2 + 2 - (\frac{2}{|J2|}) \sum_{j \in J2}(xj - 2x_2 sin(2\pi x_1 + \frac{j\pi}{n}))2$$

$$f_3 = 1 - x_2 + 2 - (\frac{2}{|J3|}) \sum_{j \in J3}(xj - 2x_2 sin(2\pi x_1 + \frac{j\pi}{n}))2$$

J1, J2, J3 are the same as in UF8

UF10:

$$f_1 = \cos\left(\frac{0.5x1\pi}{n}\right)\cos\left(\frac{0.5x2\pi}{n}\right) + 2 - \left(\frac{2}{|J1|}\right)\sum_{j \in J1}(4yj2 - \cos(8\pi yj) + 1)$$

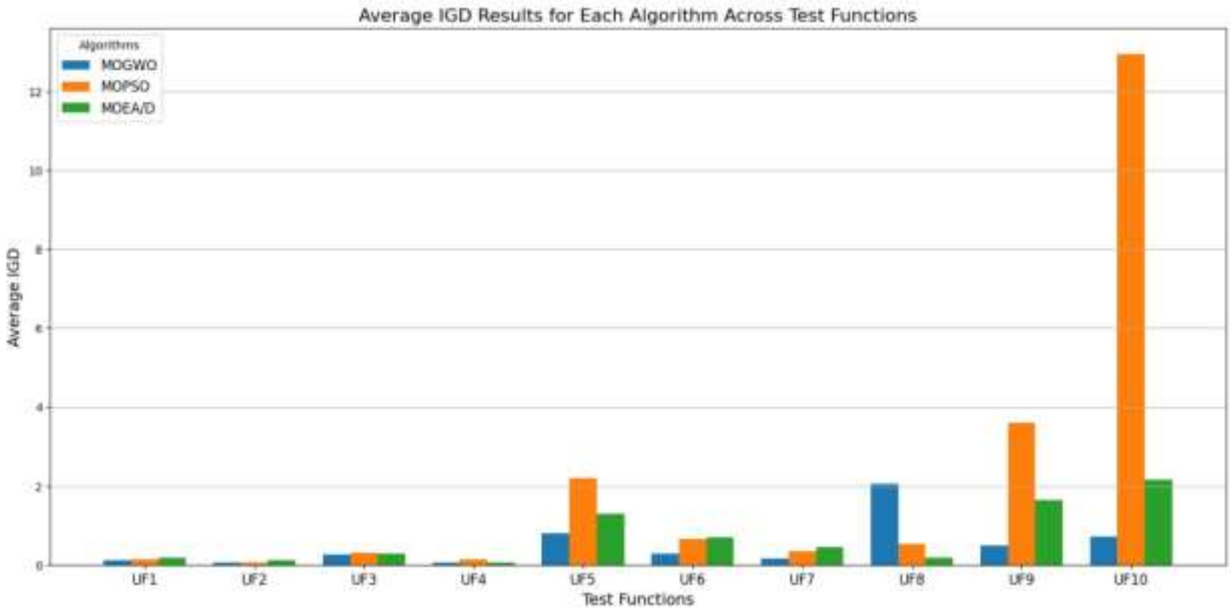
$$f_2 = \cos\left(\frac{0.5x1\pi}{n}\right)\sin\left(\frac{0.5x2\pi}{n}\right) + 2 - \left(\frac{2}{|J2|}\right)\sum_{j \in J1}(4yj2 - \cos(8\pi yj) + 1)$$

$$f_3 = sin(0.5x1\pi/n) + 2 - (\frac{2}{|J3|}) \sum_{j \in J1}(4yj2 - cos(8\pi yj) + 1)$$

$$y_j = x_j - 2x_2 sin(2\pi x_1 + \frac{j\pi}{n})$$

J1, J2, J3 are the same as in UF8

Problem	MOGWO	MOPSO	MOEA/D
UF1	0.114425	0.137005	0.187135
UF2	0.058250	0.060405	0.122343
UF3	0.255691	0.313999	0.288648
UF4	-	2.602267	2.34656
UF5	-	1.027118	1.063611
UF6	-	1.231671	2.03321
UF7	-	0.279047	0.652533
UF8	2.057772	0.536709	0.191747
UF9	0.488503	3.594533	1.637196
UF10	0.722100	12.95643	2.16220



MultObjective Optimization of a Complex System

The optimization problem involves minimizing multiple objective functions defined as follows:

- **Objective Functions**:
  - $f1(x) = \big(1 + g(xM)\big)\cos(\alpha 1\pi 2) \cdots \cos(\alpha M - 2\pi 2)$
  - $f2(x) = \big(1 + g(xM)\big)\cos(\alpha 1\pi 2) \cdots \cos(\alpha M - 2\pi 2)$
  - $f3(x) = \big(1 + g(xM)\big)\cos(\alpha 1\pi 2) \cdots \sin(\alpha M - 2\pi 2)$
  - $f3(x) = \big(1 + g(xM)\big)\cos(2\alpha 1\pi) \cdots \sin(2\alpha M - 2\pi)$
  - $\vdots$
  - $fM(x) = \big(1 + g(xM)\big)\sin(\alpha 1\pi 2)$
  - Where $g(xM) = \sum xi \in xM(xi - 0.5)^2$,

  with the constraints $0 \leq xi \leq 1$for $i = 1,2, \ldots, n$.

Parameter Suggestion:
It is recommended to set α=100This adjustment helps create a dense set of solutions in the fM- f1 plane.

Optimal Solution

The global Pareto-optimal front is achieved when xi=0.5xi=0.5 for all xi∈xMxi∈xM.

Results Table

Generation	Evaluations	Nondominated Solutions	IGD	GD
1	91	39	0.4095	0.5026
2	182	55	0.3790	0.3412
3	273	61	0.3066	0.2874
4	364	59	0.2474	0.2332

5	455	69	0.2011	0.1629
6	546	65	0.1688	0.1327
7	637	70	0.1376	0.1046
8	728	69	0.1268	0.0891
...	...	...	...	...
200	18200	91	0.0513	0.0402

Table 1: Performance metrics over generations for the optimization process, showcasing IGD and GD values.

Trend Observations:

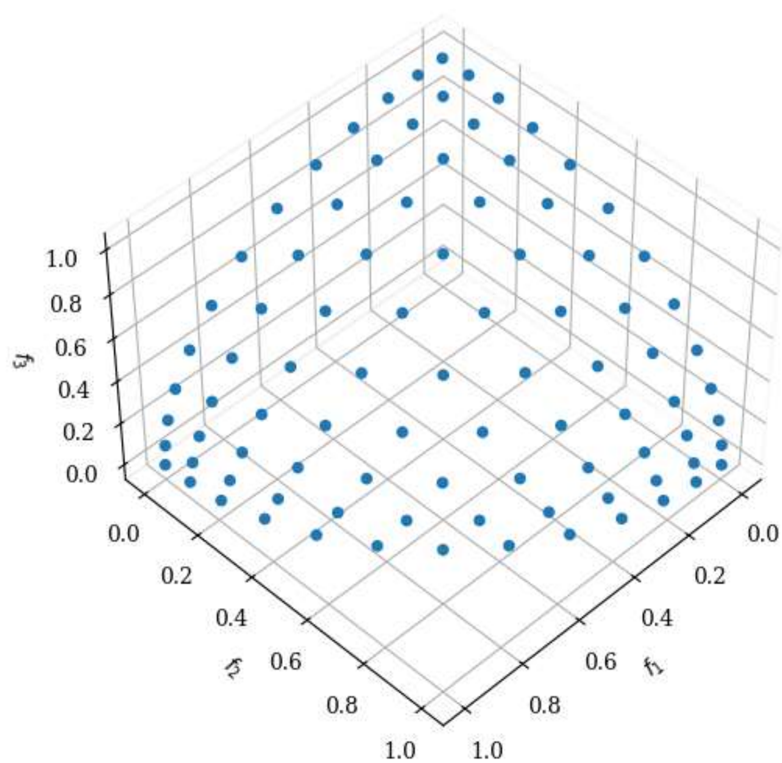
IGD (Inverted Generational Distance): The IGD values consistently decrease over generations, indicating that the solutions are getting closer to the true Pareto front. This is a positive sign of convergence.

GD (Generational Distance): The GD values also show a downward trend, suggesting that the average distance of the nondominated solutions to the Pareto front is reducing, reflecting improved solution quality.

Evaluations and Nondominated Solutions:

As the number of evaluations increases, the number of nondominated solutions tends to fluctuate but generally increases, indicating that the algorithm is effectively exploring the solution space.

By generation 200, the number of nondominated solutions stabilizes at 91, suggesting a well-defined Pareto front.



Convergence and Diversity:

The decreasing IGD and GD values indicate good convergence towards the Pareto front, while the number of nondominated solutions highlights the diversity of the solutions found.

The algorithm appears to balance exploration (finding diverse solutions) and exploitation (refining solutions towards optimality) well.

Conclusion

In conclusion, MultObjective optimization (MOO) is an essential tool for tackling complex problems across diverse fields, including economics, finance, and engineering. By employing various methodologies such as the niched-Pareto genetic algorithm and the ε-constraint method, researchers can effectively balance multiple objectives and achieve optimal solutions. The integration of Fuzzy Logic further enhances decision-making processes by accommodating uncertainty and allowing for nuanced interpretations of data through membership functions and fuzzy rules.

The exploration of the Grey Wolf Optimization (GWO) algorithm illustrates the potential of biomimicry in developing innovative optimization techniques. As we advance, the focus on creating a MultObjective Grey Wolf Optimizer that incorporates Fuzzy Optimality represents a significant step toward improving the efficacy and applicability of MOO in real-world scenarios. This research not only contributes to the theoretical framework of optimization but also aims to provide practical solutions to complex challenges, ultimately benefiting various industries and decision-making processes.

References

1. Zhang, Y., & et al. (2020). *MultObjective optimization: A review of methods and applications*. Applied Mathematics and Computation.
2. Deb, K. (2001). *MultObjective optimization using evolutionary algorithms*. Wiley.
3. Mardle, S., Pascoe, S., & Tamiz, M. (2007). *Bioeconomic models for fisheries management*. Fisheries Research.
4. Zhang, Q., & Wu, J. (2018). *Niched-Pareto genetic algorithm: A new approach for financial time series analysis*. Financial Engineering.
5. Gunasekara, A., Mehrotra, A., & Mohan, K. (2021). *Centrality measures in network models for optimizing campaign dynamics*. Journal of Political Science.
6. Khan, A., & et al. (2019). *MultObjective optimization of shell and tube heat exchangers*. Energy.
7. Zhou, Y., & Gen, M. (2001). *Global criterion method in MultObjective optimization*. Journal of Global Optimization.
8. Miettinen, K. (1999). *Nonlinear Multiobjective Optimization*. Kluwer Academic Publishers.
9. Haimes, Y. Y., & et al. (1971). *MultObjective optimization: Theory and applications*. Wiley.
10. Bard, J. F., & et al. (1998). *MultObjective optimization: Methods and applications*. Springer.
11. Zitzler, E., & et al. (2001). *Performance assessment of MultObjective optimizers: An analysis and review*. IEEE Transactions on Evolutionary Computation.
12. Zadeh, L. A. (1965). *Fuzzy sets*. Information and Control.
13. Dubois, D., & Prade, H. (1980). *Fuzzy Sets and Systems: Theory and Applications*. Academic Press.
14. Russell, S., & Norvig, P. (2020). *Artificial Intelligence: A Modern Approach*. Pearson.
15. Mamdani, E. H., & Assilian, S. (1975). *An experiment in linguistic synthesis with a fuzzy logic controller*. International Journal of Man-Machine Studies.
16. Mirjalili, S., & et al. (2014). *Grey Wolf Optimizer*. Advances in Engineering Software.