IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 20 June 2024, accepted 6 July 2024, date of publication 10 July 2024, date of current version 17 September 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3426104

RESEARCH ARTICLE

Multi-Objective Multi-Exemplar Particle Swarm Optimization Algorithm With Local Awareness

MUSTAFA SABAH NOORI[1], RATNA K. Z. SAHBUDIN[2], ADUWATI SALI[1,3], (Senior Member, IEEE), AND FAZIRULHISYAM HASHIM[1], (Member, IEEE)

[1]WiPNET Research Centre, Department of Computer and Communication Systems Engineering, Faculty of Engineering, Universiti Putra Malaysia (UPM), Serdang, Selangor 43400, Malaysia
[2]Department of Computer and Communication Systems Engineering, Faculty of Engineering, UPM, Serdang, Selangor 43400, Malaysia
[3]Institute for Mathematical Research (INSPEM), UPM, Serdang, Selangor 43400, Malaysia

Corresponding authors: Mustafa Sabah Noori (mustafasabah804@gmail.com) and Ratna K. Z. Sahbudin (ratna@upm.edu.my)

This work was supported by the Fundamental Research Grant Scheme under Grant FRGS/1/2020/TK0/UPM/02/19.

ABSTRACT Many machine learning algorithms excel at handling problems with conflicting objectives. Multi-Objective Optimization (MOO) algorithms play a crucial role in this process by enabling them to navigate these trade-offs effectively. This capability is essential for solving complex problems across diverse scientific and engineering domains, where achieving optimal solutions often requires balancing multiple objectives. One of these MOO algorithms Multi-Objective Particle Swarm Optimization (MOPSO) extends it to handle problems with multiple objectives simultaneously, but like many swarm-based algorithms, MOPSO can suffer from premature convergence or local optima solutions. Therefore, this article introduces a novel Multi-Exemplar Particle Swarm Optimization with Local Awareness (MEPSOLA) as a potent solution. The algorithm presents a developed multi-objective-aware criterion for multi-exemplar selection, adeptly balancing exploration and exploitation to avoid local optima and enhance performance across multiple objectives. It also introduces a conditional and periodic Tabu search tailored specifically for exemplar selection enhancement, improving both exploration and exploitation capabilities and avoiding premature convergence. Additionally, our method employs an improved initialization phase using equal sampling for each decision variable to ensure a comprehensive exploration of the entire solution space. A comprehensive assessment utilizing standard mathematical functions such as Fonseca-Fleming (FON), Kursawe (KUR), ZDT1, ZDT2, ZDT3, and ZDT6, and a comparison with state-of-the-art benchmarks in the field such as the Multi-Objective Evolutionary Algorithm (MOEA), Non-Dominated Sorting Genetic Algorithm (NSGA-II), and NSGA-III, validate the efficiency of MEPSOLA. Notably, MEPSOLA's solutions outperform other benchmarks in key metrics across the majority of mathematical problems, for instance in set coverage, where our method dominates other methods' solutions by 99.22%, 69%, and 93.58 %, respectively, highlighting its enhanced capability in optimizing capability within complex multi-objective optimization scenarios.

INDEX TERMS Multi-objective optimization, multi-objective particle swarm optimization, MOPSO, multi-exemplar, local search, complex optimization landscapes, Tabu search.

I. INTRODUCTION

The field of optimization has become an indispensable tool for tackling complex problems in applied mathematics and engineering. This paradigm shift recognizes that many real-world challenges can be framed as finding the ''best'' solution – the one that maximizes desired objectives while adhering to specific constraints. These desired objectives can vary widely depending on the context. For instance, it might seek to maximize performance, accuracy, or coverage, while minimizing resource usage, error rates, delay, or even the cost-effectiveness [1]. The impact of optimization extends far beyond theoretical exercises, with applications now driving advancements in diverse fields like security [2], energy

The associate editor coordinating the review of this manuscript and approving it for publication was Sotirios Goudos.

management [3], financial modeling [4], transportation logistics [5], data analysis [6], and even medical field [7].

In Artificial Intelligence (AI) and Machine Learning (ML), optimization algorithms are essential [8]. These algorithms, crucial for constructing optimization models and identifying optimal values or parameters from the objective function [8]. There are two main types of optimization techniques used in various fields: single-objective and Multi-Objective Optimization (MOO). In MOO involves optimizing multiple conflicting objectives simultaneously, often under conflicting constraints. Unlike single-objective optimization, which seeks a single optimal solution, MOO aims to find a set of solutions known as the Pareto front. These solutions represent trade-offs among the objectives, where no solution in the set is superior in all objectives. This approach acknowledges the complexity and multi-faceted nature of real-world problems, where balancing different objectives is necessary for optimal decision-making [9].

One of the standout methods in MOO is the Multi-Objective Particle Swarm Optimization (MOPSO), an advancement of the Particle Swarm Optimization (PSO) algorithm [10], inspired by the social behaviors of birds flocking [11]. MOPSO extends PSO principles to solve multiple objectives, directing a swarm of particles or solutions into the search space. Each particle modifies its path based on personal and neighboring experiences, aiming for efficient exploration. MOPSO's strength lies in its simplicity and ability to quickly converge to a set of diverse solutions [12].

Complex solution spaces where MOPSO algorithms like MEPSOLA excel involve various challenging conditions: multiple conflicting objectives requiring trade-offs, as seen in engineering and design; high dimensionality, common in large-scale feature selection [2] and financial optimizations [4]; non-linear objective functions that create multiple local optima, typical in network design; discrete and combinatorial decision variables in scheduling and routing tasks; dynamic environments like inventory management where conditions change over time [13]; multi-modality, where distinguishing between multiple satisfactory solutions is difficult, seen in ecological modeling; constraint-rich problems with numerous complex constraints, prevalent in aerospace industries; and environments with uncertainty or noise, such as in simulation-based optimizations.

MOPSO has proven to be a valuable tool in tackling complex optimization problems involving multiple objectives. Researchers have developed and refined MOPSO to address various scenarios [14], aiming to achieve a balance between exploration (discovering new possibilities) and exploitation (refining promising areas) [15]. This ongoing refinement helps MOPSO avoid premature convergence on suboptimal solutions [16]. However, the literature still lacks a comprehensive MOPSO algorithm that integrates conditional local searching for exemplar selection, multi-exemplar approaches, and smart initialization to ensure a well-distributed initial population, which could further boost the algorithm's exploratory abilities.

In MOPSO, exemplars are crucial reference points that guide the swarm's search by directing particles towards promising solution space. Exemplars selection, usually preformed based on performance of particle or certain equation, significantly influences search efficiency and convergence [17]. MOPSO, exemplars are often chosen from a repository of non-dominated solutions, ensuring that the swarm is guided by the most effective solutions discovered so far [18]. Therefore, the concept of multi-exemplar usage introduces an additional layer of diversity to the search process. Instead of each particle following a single exemplar, the multi-exemplar approach allows particles to be influenced by multiple guidance points [19]. This diversity is crucial in avoiding premature convergence and ensuring a thorough exploration of the solution space. By having multiple exemplars, particles are exposed to a wider range of solution characteristics, encouraging them to explore different regions and thereby enhancing the algorithm's ability to discover a diverse and well-distributed Pareto front. The multi-exemplar strategy balances the exploration and exploitation phases of the search, making it highly effective in complex MOO scenarios. However, relying solely on multiple exemplars might not fully explore complex MOO scenario due to the risk of particles becoming trapped in local optima. Hence, incorporating local search methods is essential to mitigate this issue, ensuring a more thorough exploration and improving the algorithm's overall performance in navigating challenging optimization landscapes [10].

Local search methods, such as Tabu search (TS), are designed to intensively explore the neighborhood of a solution, providing a deeper insight into local regions of the solution space [20]. When integrating local search algorithms like TS, Simulated Annealing (SA), or Variable Neighborhood Search (VNS) with MOPSO methods, TS often emerges as a particularly effective choice. Unlike SA, which relies on probabilistic decisions influenced by a cooling schedule that can demand complicated adjustments, TS employs a memory-driven approach that systematically avoids revisiting previous solutions, offering a more consistent and stable search path. This deterministic approach is crucial for structured exploration in multi-objective optimization, where managing complex objectives systematically is paramount. Moreover, TS's adaptability through flexible strategies such as varying the Tabu list length and utilizing aspiration criteria enables more controlled exploration and exploitation. This complements the global search capabilities of MOPSO effectively, ensuring seamless integration that leverages historical data to optimize decision-making processes. In contrast, other local search methods like VNS offer multiple neighborhood structures but may not provide the same level of strategic depth in avoiding previously explored areas, which is critical in complex multi-objective settings. Therefore, TS is often preferred for its predictability, control, and the ability to handle complex multi-objective scenarios efficiently, making it a robust choice in structured and reliable optimization processes [21].

The combination of temporary multi-exemplar searching with local search methods could potentially address the limitations of both strategies. While the multi-exemplar approach improves global exploration and diversity, local search methods can efficiently exploit the local regions, leading to a more comprehensive search. However, such an integrated approach is not commonly found in existing literature. The lack of this integration results in a missed opportunity for enhancing the algorithm's capability to navigate complex optimization problems effectively. Without local search, particles may overlook finer details in the solution space, while without multi-exemplar guidance, they may get trapped in local optima. An ideal algorithm would dynamically switch between these two modes, employing multi-exemplar searching for global exploration and local search methods for detailed exploitation. This synergy could lead to more robust and effective optimization results, especially in problems where the solution space is rugged or has many competing objectives.

This article introduces an innovative MOO algorithm, grounded in the theoretical framework of PSO, and provides the following contributions:

1. To our understanding, this research presents a MOPSO algorithm specifically designed to address the complexities of solution spaces. The algorithm not only incorporates but also refines multiple techniques to efficiently manage the multi-objective nature.
2. A multi-objective-aware criterion for multi-exemplar selection is also presented. This criterion adeptly balances between the dual imperatives of exploration and exploitation, thereby optimizing the algorithm's performance across multiple objectives.
3. Enabling of Tabu search for exemplar selection only and for every T period provides both global and local awareness in the searching behavior and enable avoidance of local minima.
4. Rigorous evaluation of the algorithm has been conducted using standard mathematical benchmarks prevalent in the optimization domain. Furthermore, the algorithm's performance metrics have been assessed using established multi-objective evaluation standards, and comparative analyses have been performed against benchmark algorithms in the field of optimization.

The remainder of the article is organized as follows: Section II provides a literature survey and related works. Section III presents our developed methodology. Section IV discusses evaluation metrics and benchmarking functions. Section V explains the experimental design and the results analysis and discussion. Finally, Section VI represents the conclusion and directions for future work.

II. LITERATURE SURVEY

PSO is a computational technique that has gained prominence for its ability to solve complex optimization problems. PSO, inspired by social behaviors of bird flocking, excels in complex optimization tasks. Its adaptability lies in solution representation and particle mobility, offering simplicity with minimal parameter adjustments [10]. Each particle in PSO represents a potential solution, with position indicating solution quality [22]. This representation is adaptable to various problem types, including continuous, discrete, and combinatorial [14]. Particle mobility, determined by velocity updates based on personal and neighbor's best positions, facilitates a balance between exploration and exploitation [23], [24].

MOPSO extends PSO to handle multi conflicting objectives. Key features of MOPSO include a repository for storing Pareto-optimal solutions and a grid mechanism ensuring solution diversity [25], [26]. Recent advancements include variable length PSO [27] and autonomous Convolutional Neural Network (CNN) architecture optimization using an evolved encoding method and ''Disabled layer'' concept for variable-length solutions [28]. Moreover, a study combined PSO with evolutionary game theory, introducing Modified Self-Adaptive PSO (MSAPSO) for parameter optimization, balanced exploration and exploitation, and convergence analysis procedure [29]. Another integration of enhanced PSO with a Pareto archive addresses multi-objective reactive power optimization, employing a greedy strategy for intensive local searches [15]. A study introduce method named a PSOBSA, a combine of PSO with Search Optimization Algorithm (BSA), aims to overcome PSO's diversity loss and local optima entrapment [1].

Further advancement for MOPSO with a fuzzy dominance relationship and crowding distance measure are proposed, with a tolerance coefficient ensuring decision-maker satisfaction [30]. In the term of addressing PSO's complexity, a simplified MOPSO algorithm is developed, compared qualitatively with real-coded elitist non-dominated sorting genetic algorithm [31]. To overcome limitations in archive size and diversity maintenance, MOO based PSO with an archive updating mechanism using nearest neighbor strategies is introduced [32]. Furthermore, a multi-modal MOO-based PSO with a self-adjusting strategy tackles convergence deterioration, employing sub-swarms for solution diversity [33]. The impact of swarm connection topology in MOPSO is analyzed, enhancing sensitivity to swarm topologies using regular graphs [34]. An improved guidance for swarms is proposed by pairing with suitable leaders, adapting strategies for leader selection based on the evolutionary state and dominance relationship [35].

Moreover, dynamic MOO solutions address shifting objective importance, employing a double search strategy for PSO and a piecewise strategy for rapid convergence [13], [36]. High-dimensional challenges are tackled by a two-stage optimizer, population cooperation based PSO, utilizing auxiliary and sub-population cooperation for convergence rate and diversity balance [37]. A two-archive MOPSO focuses on convergence and diversity, with archives refined using indicator-based and density estimation approaches, and genetic techniques for solution quality enhancement [38]. Lastly, a novel global best solution selection method, based

on virtual generational distance, optimizes both convergence and diversity, complemented by an adaptive personal best solution selection based on evolutionary stages, strengthening exploitation or exploration [39]. In [40], the authors tackle the issue of weak selection pressure in MOPSOs, which can hinder the generation of robust Pareto solutions in many-objective problems. They introduce a Hybrid Global Leader Selection Strategy (HGLSS) that features two distinct global leader selection mechanisms aimed at enhancing both exploration and exploitation capabilities. Each particle, representing a potential solution, can select between these two mechanisms to identify its global best leader. The approach also incorporates an external archive, utilized for maintaining solution diversity and updated through Pareto dominance and density estimation. This strategy has shown promising results, outperforming other state-of-the-art multi-objective metaheuristics in terms of the Modified Inverted Generational Distance (IGD+) indicator, providing a significant advancement in the field of MOPSO.

Upon reviewing the literature in the Table 1, for the recent MOPSO methods developments it became noticeable, uncovers a significant gap, the lack of a comprehensive method that joining multi-exemplar strategies, robust local searching mechanisms for exemplar selection enhancement and an improved initialization phase. Current PSO-based MOO approaches primarily focus on exploring and exploiting trade-offs and improving the selecting diverse solutions. However, they often overlook the integration of smart initialization and underutilize local searching techniques and multi-exemplar strategies.

This article, proposes a method called Multi-Exemplar Particle Swarm Optimization with Local Awareness (MEPSOLA), which addresses the literature gap by incorporating smart initialization, advanced solution selection-based grid selection, and an innovative combination of multi-exemplar strategies with a global and conditional local search approach. This integration promises a more holistic exploration of the solution space, potentially leading to more diverse and optimal solutions, enhancing both exploration and convergence while avoiding premature convergence traps. Thus, our approach aims to enhance the effectiveness of Multi-Objective Optimization (MOO) in complex optimization scenarios.

III. METHODOLOGY

This section details the methodology of the MEPSOLA algorithm, it begins with problem formulation, followed by describing the general flowchart. Next, the introduction of the initial phase of solution initialization is provided. Then demonstration of solutions' mobility and interactions in the search space is introduced, which is divided into three parts: solutions interactions with exemplars, solutions mutation, and exemplar selection. Next, local searching based on Tabu search is presented, followed by a detailed explanation of the main algorithm. Lastly, the exploration mechanisms are presented.

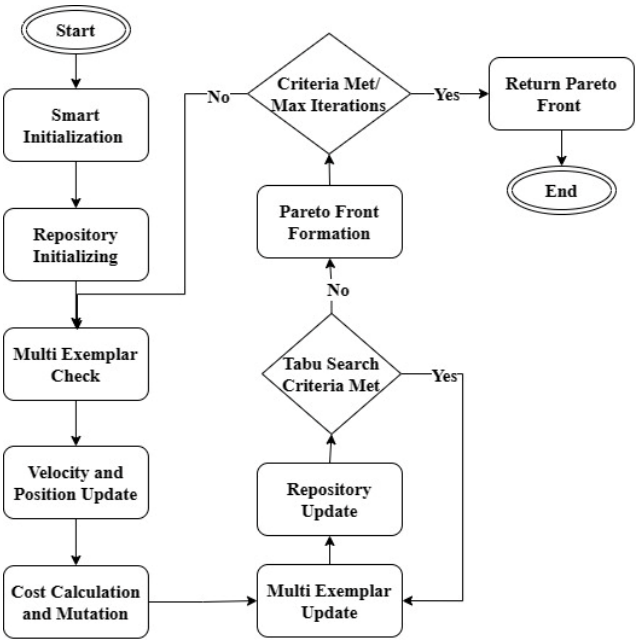

FIGURE 1. Flowchart of the general process for MEPSOLA.

A. PROBLEM FORMULATION

MOO is an essential tool in various fields, enabling the simultaneous optimization of multiple conflicting objectives. The challenge of identifying optimal solutions under complex constraints, illuminating the complexities of decision-making in environments where trade-offs are inevitable.

Let's assume that $f = (f_1, f_2, \ldots, f_n)$, be a vector of n objective functions to be optimized. Each objective function $f_i{:}X \rightarrow \mathbb{R}$ maps a solution $x \in X$ to a real number where X is the set of all feasible solutions defined by a set of constrains $g_j(X)$. $X = \{x \in \mathbb{R} | g_j(X) \leq 0, h_k = 0, \} j = 1, \ldots m$, $k = 1, 2 \ldots p$ where m denotes the number of inequality constraint $g_j(X) \leq 0$, p denotes the number of equality constraints $h_k = 0$.

A solution x_1 is said to dominate another solution x_2 or $x_1 \prec x_2$ if and only if: $f(x_1) \leq f(x_2)$ for all $i = 1, 2 \ldots n$ and $f(x_1) < f(x_2)$ for at least one $1 \leq i \leq n$.

The multi-objective optimization problem based on the concept of non-domination can be formally stated as: Find $x^* \in X$ such that $\nexists x \in X$ for which $x \prec x^*$. All the solution x^* that meet this condition are given in the Pareto front $X^* = \{x \in X | \nexists x' \in X, x' \prec x\}$.

The ultimate aim is to determine an optimal set of solutions, each offering a unique balance among the different goals, known as the best compromise solutions.

B. GENERAL FLOWCHART FOR MEPSOLA

This section presents the general flowchart in Fig. 1, illustrates the MEPSOLA algorithm procedure, which follows a systematic optimization sequence. It begins with smart initialization to strategically set up the population and repository. The algorithm iterates through cycles of exemplar

TABLE 1. Literature for existing methods of multi-objective particle swarm optimization.

Ref	MOO Method	Smart Initialization	Trade-off Exploration & Exploitation	Non-Dominated Solution	Solution Selection	Local Searching	Multi-exemplar
[29]	Modified Self-Adaptive PSO	×	√	√	Roulette wheel / density estimation	×	×
[15]	Particle Swarm Optimization and Pareto Archive	×	√	√	Minimum sum between objective function and optimal solution	×	×
[1]	Particle Swarm Optimization & search optimization algorithm	×	√	×	Greedy selection	√	×
[28]	Particle Swarm Optimization & variable-length CNN	×	×	×	Global Best Selection and Decoding	×	×
[30]	Fuzzy multi objective Particle Swarm Optimization	×	√	√	Fuzzy crowding distance / tolerance coefficient (λ)	×	×
[31]	Multi-objective Particle Swarm Optimization	×	√	√	Elitism operator	×	×
[32]	PSO with archive updating mechanism	×	√	√	Distance	×	×
[33]	Multi multi-objective particle swarm Self-Adjusting Strategy	√	√	√	Crowding distance/ dominance rank	×	×
[34]	Speed-constrained Multi-objective Particle Swarm Optimizer	×	√	√	Crowding Distance	×	×
[35]	PSO/ adaptive multiple selection strategy	×	√	√	Sparse degree and rank	×	×
[41]	Cooperative coevolutionary multi-guide particle swarm optimization	×	√	√	Crowding distance-based archive	×	×
[13]	Dynamic multi-objective optimization problems PSO	×	√	√	Crowding distance-based ranking	×	×
[37]	Population Cooperation based Particle Swarm Optimization	√	√	√	Crowding distance	×	×
[38]	Multi-objective particle swarm optimization algorithm based on two-archive mechanism	×	√	√	Euclidean distance	×	×
[39]	Multi-objective particle swarm optimization	×	√	√	Roulette wheel selection	×	×
[36]	Dynamic Pareto bi-level Multi-Objective Particle Swarm Optimization	×	√	√	Crowding distance and the Pareto ranking	×	×
[40]	Multi-objective particle swarm optimizer with Hybrid Global Leader Selection Strategy (HGLSS)	×	√	√	Pareto dominance and density estimation	×	√
Proposed method	Multi-Exemplar PSO with Local Awareness	√	√	√	Grid based selection	√	√

validation, particle velocity and position updates, and detailed cost computation with mutation. A notable feature is the inclusion of a conditional Tabu search for local exemplar selection enhancement. This iterative process continues until convergence criteria or maximum iterations are reached, culminating in the formation of a Pareto front that indicates the optimal set of solutions. The detailed methodology of MEPSOLA will be thoroughly described in the subsequent sections of the article.

C. SOLUTIONS INITIALIZATION

The initialization stage is a crucial step is undertaken where for each decision variable, equal sampling within its range is conducted. This process is pivotal as it ensures that the initial values for the decision variables are selected in a manner that is representative of their entire range. The methodology here involves assigning *Range_values* for each decision variable. This equitable distribution of initial values is fundamental for the algorithm's performance, as it guarantees a comprehensive exploration of the solution space right from the outset. Without this balanced initialization, the algorithm could potentially overlook significant areas of the solution space, leading to suboptimal outcomes.

The Cartesian product is applied to these *Range_values*, resulting in a set of permutations. These permutations embody all possible combinations of decision variable values within their respective ranges. The generation of these permutations is a testament to the thoroughness with which the solution space is being explored.

The population is then established by randomly selecting a subset of these permutations, the size of which is determined by *pop_size*. Each particle in this initial population is assigned a position from Positions. Subsequently, the initialization of the particle's velocity and the calculation of its cost are performed. This random sampling from a comprehensively generated permutation set ensures that the initial population is not biased towards any specific region of the solution space.

The formation of the Repository involves the identification of non-dominated particles from the initial population. This process, executed through the function *get_particles_non_dominated*(*population*), is essential for maintaining a diverse set of solutions that are not overshadowed by any dominant trait or characteristic. Each particle in the population is assigned an exemplar through the *exemplar_selection*() function. This step is integral to guiding the search process in the algorithm, as exemplars serve as reference points for particles to follow.

Finally, the algorithm concludes by returning the population alongside the Repository. The initial population, having been generated through a process that emphasizes fairness and thoroughness in sampling, sets a solid foundation for the subsequent computational steps. This fair initialization is not merely a procedural necessity but a cornerstone that significantly influences the algorithm's efficiency and effectiveness in exploring the solution space and converging towards optimal solutions, the process of initialization phase illustrated in Algorithm 1.

Algorithm 1 Pseudocode of generate of first population

Input:
- Pop_size: The size of the population.

Output:
-First_pop: The first generated population.
- Repository: A repository of non-dominated particles.

Procedure:

```
    Initialization:
    For each decision variable
      Range_values(decision variable) = do equal
    sampling with the range of decision variable
     End
    Permutations=apply Cartesian product on
    Range_values
    Positions = random.sample(permutations, pop_size)
Population Setup:
  - For each particle in particles:
    - Assign a position from Positions to
Particle.position.
    - Initialize Particle.velocity.
    - Calculate Particle.cost.
Repository Formation:
    - Form Repository by identifying non-dominated
particles from the population using
get_particles_non_dominated(population).
Exemplar Assignment:
    - For each particle in particles:
    - Assign an exemplar to Particle.exemplar by calling the
exemplar_selection() function.
```

Return:
- Return population and Repository.

The improved initialization phase is pivotal for the success of the MEPSOLA algorithm in handling multi-objective problems. By establishing a solid foundation of diverse and representative initial solutions, this phase significantly influences the quality of solutions and the speed at which the algorithm converges. These enhancements not only streamline the computational process but also lead to comprehensively covering the entire search space, preventing initial clustering of solutions and improving the diversity of the initial population [42].

D. SOLUTIONS MOBILITY AND INTERACTIONS

This subsection presents three types of operations related to solutions' mobility. Firstly, solutions interaction with exemplar. Secondly, solution conditional mutation. Lastly, exemplar selection.

1) EXEMPLAR AND SOLUTIONS INTERACTION

In optimization algorithms, particularly those similar to PSO, the nuanced interaction between a solution and its exemplar is

defined by a specific mathematical formula as in Equation 1 [27].

$$v_i^{t+1} = w \cdot v_i^t + c \cdot r_i \cdot \left(p^t_{exmplr(i)} - x_i^t\right) \quad (1)$$

where v_i^t denotes the current velocity of the solution.

r_i denotes a random number between 0 and 1.

$p^t_{exmplr(i)}$ denotes the position of the exemplar of dimension d for solution i.

The formula involves multiple components that collectively determine the movement of a solution within the search space. At its core, this interaction involves the update of a solution's velocity, a process influenced by the solution's current velocity, represented by v_i^t. This element of velocity is crucial as it dictates the rate and direction of the solution's movement, fundamentally shaping its direction through the search space.

The influence of inertia weight, denoted as w, plays a pivotal role in this process. It determines how much of the solution's previous momentum is carried forward into its new velocity, thus balancing the forces of exploration and exploitation within the algorithm. A larger inertia weight encourages a broader, more explorative approach, while a smaller weight focuses on localized, detailed searches.

The interaction is further characterized by a randomized attraction towards the exemplar, represented by $p^t_{exmplr(i)}$. This exemplar, often the best solution found by either the individual or the entire swarm, serves as a point of attraction. The level of this attraction is influenced by a random factor, r_i, a number generated between 0 and 1, introducing unpredictability and variation in the movement of the solution. This randomness is critical in preventing the solution from becoming trapped in local optima.

An acceleration coefficient c, modifies this attraction, steering the solution more decisively towards promising regions within the solution space. The influence of this coefficient is crucial in dictating the interaction of the solution's movement towards the exemplar.

Finally, the interaction takes into account the solution's current position, x_i^t, by subtracting it from the exemplar's position. This difference is integral as it quantifies the relative position of the solution to its guiding point, thus directing the update of velocity.

Through this interaction, each solution in the population is methodically guided towards optimal or near-optimal regions. The exemplar acts as a point, directing the solution's journey, while the inertia and random elements ensure a diverse and comprehensive exploration of the solution space. This balance of individual movement influenced by personal and collective experiences encapsulates the essence of population-based optimization techniques, leading to effective and efficient discovery of solutions.

2) SOLUTION MUTATION

The mutation technique is a concept borrowed from evolutionary algorithms. It serves as a mechanism to introduce diversity into the population. While not a standard component of PSO, mutation can be incorporated in particle swarm optimization to prevent premature convergence. In our proposed algorithm, we adopt picking randomly one of the decision variables and replacing its value with another from its originally generated range values.

3) EXEMPLAR SELECTION

Multiple exemplar strategies in advanced MOPSO methods, such as MEPSOLA, significantly enhance the optimization process compared to traditional methods that typically rely on a single exemplar for guidance. These strategies introduce greater exemplar diversity by allowing particles to follow multiple exemplars, each representing different regions of the search space, which promotes a broader and more diverse exploration of the solution space. Also, the dynamic exemplar selection enables particles to switch between exemplars based on their need for exploration or exploitation, adapting to the evolving solution landscape. This approach combines local and global guidance, preventing premature convergence on certain parts of the Pareto front and encouraging exploration of less dominant regions [19].

Algorithm 2 Exemplar Selection Procedure

Input:
- particle: The particle for which the exemplar is to be selected.

Output:
- Exemplar: The selected exemplar for the input particle.

Procedure:
Initialization:
- Set a Threshold_repetition value.
- Initialize Used_exemplars as an empty directory.

Selection of Exemplar:
- Choose a random Exemplar from the Repository.
- **If** Exemplar is not in Used_exemplars :
  - Add Exemplar to Used_exemplars and set its count to 1.
- **Else**, if Used_exemplars[Exemplar] is less than Threshold_repetition:
  - Increment the count of Exemplar in Used_exemplars.
- **Otherwise**:
  - Select a new Exemplar by applying tabu_search(particle.position) **Algorithm 3**

Return:
- Return the selected Exemplar.

End

In the exemplar selection phase, as illustrated in Algorithm 2, this process is critical for guiding the search behavior of particles in an optimization algorithm. The conditional calling of local search, the temporary usage of an exemplar, and the utilization of multi-exemplar are key elements that significantly influence the algorithm's effectiveness and efficiency. The algorithm begins with an

initialization step, where a *Threshold_repetition* value is set. This threshold is crucial as it defines the maximum frequency with which an exemplar can be used before triggering alternative search mechanisms. Additionally, *Used_exemplars* is initialized as an empty directory. This directory will track the usage frequency of each exemplar, thereby playing a vital role in managing the diversity of the search process.

The primary step involves choosing a random Exemplar from the Repository. The randomness here is vital for ensuring an unbiased selection process. The selection process then checks whether the chosen Exemplar is already in *Used_exemplars*. If it is not, the Exemplar is added to *Used_exemplars* directory, with its count set to 1. This step ensures that each Exemplar gets a chance to influence the search process. If the Exemplar is already used but its count is less than *Threshold_repetition*, its count is incremented. This mechanism allows for the repeated use of effective exemplars but within a controlled limit. The role of conditional calling of local search is highlighted when the count of an Exemplar reaches the threshold. In this case, instead of continuing to use the same Exemplar, the algorithm calls Algorithm 3, which is responsible on Tabu search. This shift to a local search method, triggered by the threshold, ensures that the search does not stagnate due to over-reliance on a particular Exemplar. It introduces a new dimension to the search process, allowing for exploration in different regions of the solution space. The temporary usage of an exemplar, as governed by the *Threshold_repetition* and the use of *Used_exemplars*, ensures that no single exemplar dominates the search process. This temporary usage allows for a dynamic and adaptive search strategy, where exemplars are rotated based on their efficacy and frequency of use. It prevents premature convergence and maintains the diversity of the search process. The utilization of multiple exemplars, as facilitated by the dynamic selection process, plays a pivotal role in exploring various regions of the solution space. By allowing particles to be influenced by different exemplars over time, the algorithm benefits from a multi-faceted search approach. This diversity in guidance helps in effectively navigating the solution landscape, aiding in the avoidance of local optima and enhancing the probability of finding global optima. Finally, the algorithm returns the selected Exemplar, which is then used to guide the movement of the particle in the optimization process.

E. TABU SEARCH

Tabu Search algorithms as a conditional interaction is employed through a series of steps to navigate and enhance the exemplar selection process within the optimization problem as in Algorithm 3. Its only enabled when the *Threshold_repetition* reach the maximum counts which is the maximum frequency with which an exemplar can be used. The process started by setting the *initial_solution* as both the *current_solution* and the *best_solution*. This dual assignment signifies the commencement of the search process from a defined point, with the *best_solution* being a placeholder for the optimal solution encountered during the search. A *tabu_list* is also introduced, initially as an empty list. The function of this list is to record the solutions that have been recently explored. The maintenance of this list is critical as it helps in avoiding repetitive exploration of the same solutions, thereby preventing the algorithm from being trapped in cyclic patterns.

Another key component is the *tabu_list_max_size*, a predefined value that limits the number of entries in the *tabulist*. This constraint ensures that the list only retains a certain number of recent solutions, balancing memory efficiency with the effectiveness of avoiding recent solutions.

The core of the algorithm is a loop that continues until a specified stopping criterion is met. Within this loop, the algorithm generates a neighborhood of solutions surrounding and close to the *current_solution*. These solutions typically representing slight modifications or variations of it. From this neighborhood, the algorithm selects the *best_neighbor*, which is the most promising solution that is not in the *tabulist*. This step is crucial as it guides the search towards areas of the solution space that are yet unexplored or hold potential for better solutions.

Upon identifying the *best_neighbor*, the algorithm updates the *current_solution* to this new solution. Concurrently, it checks if this *best_neighbor* is an improvement over the *best_solution* found so far. If it is, the *best_solution* is updated to reflect this new, better solution.

Finally, the *tabu_list* is updated with the *best_neighbor*. This update involves adding the new solution to the list and ensuring that the list size does not exceed the predefined *tabu_list_max_size*. This management of the tabu list is essential for the algorithm's ability to efficiently navigate through the solution space without revisiting recently considered solutions. The algorithm concludes by returning the *best_solution* once the stopping criteria are met. This solution is the optimal solution found by the algorithm in its exploration of the solution space, guided by the strategic avoidance of recently explored solutions through the tabu list.

F. MULTI-EXEMPLAR PARTICLE SWARM OPTIMIZATION WITH LOCAL AWARENESS (MEPSOLA) GENERAL PROCEDURE

The algorithm initiates by setting the initial population of particles, *First_population*, and a Repository for storing non-dominated particles. The *First_population* is derived from Algorithm 1, which emphasizes fair and comprehensive initialization, ensuring a diverse and representative starting point for the optimization process. The algorithm in this phase assign exemplars for each particle from the current set of non-dominated solutions found in the population.

Then, in the main loop of the algorithm, which repeats until predefined stopping criteria are met, each particle in the population undergoes a series of steps. An Exemplar Check is first conducted for each particle. If a particle's position is identical to its exemplar's, the exemplar is adjusted

Algorithm 3 Tabu Search for Exemplar Selection

Input:
initial_solution
Output:
best_solution
Start:
current_solution = initial_solution
best_solution = initial_solution
tabu_list = empty list
tabu_list_max_size = some predefined value
while stopping_criteria_not_met:
    neighborhood = generate_neighbors(current_solution)
    best_neighbor = find_best_candidate(neighborhood, tabu_list)
    current_solution = best_neighbor
    if better_than(best_neighbor, best_solution):
        best_solution = best_neighbor
    update_tabu_list(tabu_list, best_neighbor, tabu_list_max_size)
    return best_solution
End

using the *get_close_number* function. This step is crucial as it maintains diversity and avoids stagnation in the search process. Subsequently, the particle's velocity and position are updated to reflect the ongoing dynamics of the algorithm. These updates play a crucial role in navigating the solution space and are influenced by the guidance provided by the exemplar. Following this, each particle's cost is calculated, and a mutation is potentially applied, introducing variability and aiding in exploring new areas of the solution space. The cost calculation is vital for evaluating each solution's quality.

If a particle's cost is better than that of its exemplar, the exemplar is updated using the *exemplar_selection* method from Algorithm 2. This method selects exemplars in a manner that balances exploration and exploitation, using a tabu list to avoid repetitive exploration and enhance the search's diversity. After processing all particles, the *Repository* is updated with non-dominated particles from the population. This update is a crucial step, as it maintains a set of optimal solutions found so far, influenced by the diverse and dynamic exemplars. The algorithm concludes by forming the *Pareto_front* from the non-dominated solutions in both the repository and the population, as determined by their costs and positions. This front represents the set of optimal solutions, balancing various objectives. The final step is returning the Pareto front, marking the end of the optimization process.

G. EXPLORATION MECHANISMS

The exploration behavior of our proposed algorithm is resulted from several mechanisms that can be stated as follows.

Algorithm 4 Multi-Exemplar Multi-Objective Particle Swarm Optimization

Input:
First_population : Initial set of particles.
Repository : Storage of non-dominated particles.
Output:
Pareto_front The final set of optimal solutions.
Procedure:
Initialization using **Algorithm 1**
- Set population to First_population.
- Initialize repository as Repository.
-initialization of exemplar
Main Loop:
- While the stopping criteria are not met:
- For each particle in population:
- Exemplar Check:
- **If** particle.position is equal to particle.exemplar :
- Adjust particle.exemplar using get_close_number(particle.exemplar).
- Velocity and Position Update:
- Update particle.velocity using Equation(1)
- Update particle.position using $x_i^{t+1} = x_i^t + v_i^{t+1}$
- Cost Calculation and Mutation:
- Calculate particle.cost.
- Apply mutation to the particle with apply_mutation(particle).
- Exemplar Update:
- **If** particle.cost is better than particle.exemplar.cost :
- Update particle.exemplar using exemplar_selection() using Algorithm 2
- Repository Update:
- Update repository with non-dominated particles from population using get_particles_non_dominated(population).
Pareto Front Formation:
- Form Pareto_front from the non-dominated solutions in repository and population using get_solutions_non_dominated(repository, population).
Return:
- Return Pareto_front.
End

1. The incorporation of smart or equal initialization to enable fair sampling of initial population in the objective space. This ensures a diverse starting point for the optimization process, laying the groundwork for comprehensive exploration.
2. The incorporation of Tabu search to be conducted on each exemplar every T period. This integration enhances local exemplar selection, preventing Local optimal solutions and encouraging exploration in diverse regions of the solution space.

3. The incorporation of mutation which is an evolutionary mechanism. This mechanism, not typical in traditional PSO, helps prevent premature convergence by exploring new areas of the solution space.
4. The incorporation of grid aware solutions selection mechanism for selecting solutions as a part of the repository. This approach ensures that solutions are strategically chosen to maintain diversity and representativeness in the repository, further enhancing the exploration capabilities of the algorithm.

IV. EVALUATION

The evaluation section comprises two primary subsections: first, it includes the benchmark mathematical functions utilized to evaluate the algorithms. Second, presents the evaluation measures employed to evaluate the algorithm performance and compare them to other benchmarks.

A. BENCHMARKING MATHEMATICAL FUNCTIONS

The main aim of MOO algorithms is to find a set of solutions that represent the best possible trade-offs between conflicting objectives in a given problem. In the evaluation of MOO algorithms, several different mathematical problem functions are utilized to evaluate the algorithm's performance and compare it against other state-of-the-art benchmarks. These problems vary in shapes and complexity and are commonly used to evaluate MOO algorithms. The functions include FON, KUR, ZDT1, ZDT2, ZDT3, and ZDT6 [43]. The details of these mathematical problems are described in Table 2.

These mathematical functions utilized in this study encompass a diverse range of complexities and shapes, each posing unique challenges for optimization algorithms. The functions range from FON's non-convexity to KUR's disconnected Pareto shape, and from ZDT1's convexity to ZDT6's non-convex and non-uniform Pareto shape. Collectively, these functions serve as robust challenges for evaluating MEPSOLA's efficacy across various optimization problems.

B. EVALUATION MEASURES

In the MEPSOLA evaluation process, several measures are used to assess it performance and compare to other benchmarks of MOEA, NSGA-II, and NSGA-III. These evaluation measures serve to measure various aspects of the algorithm's behavior and solution quality. These evaluation measures include, set coverage, hyper-volume, delta metric, generational distance, and NDS.

1) SET COVERAGE MEASURE

The Set Coverage measure, also known as the C-metric, measure the proportion of solutions in one Pareto set that are dominated by another. It compares two Pareto sets, Ps1 and Ps2, as shown in Equation 2.

$$C\left(Ps1;Ps2\right) = \frac{\left|\{y \in Ps2 \mid \exists x \in Ps1 : x \prec y\}\right|}{|Ps2|} \tag{2}$$

where C represents the ratio of non-dominated solutions in Ps2 that are dominated by non-dominated solutions in Ps1, to the number of solutions in Ps2.

2) HYPER-VOLUME MEASURE

The HV-metric is a widely used measure in MOO algorithms for performance evaluation. It calculates the volume of the dominated portion of the objective space relative to a reference point. This region comprises the union of hypercubes with diagonals equal to the distance between the reference point and a solution x from the Pareto set PS. Higher values of this measure indicate more desirable solutions. HV calculated using Equation 3.

$$HV = volume\left(\bigcup_{x \in Ps} HyperCube\left(x\right)\right) \tag{3}$$

3) DELTA MEASURE (Δ)

The Delta measure is a metric used to evaluate the distribution and spread of the non-dominated solutions in MOO. The idea behind the delta measure is to ensure that the solutions obtained are not only non-dominated but also uniformly distributed across the Pareto front. To calculate the delta measure according to the Equation 4.

$$\Delta = \frac{(df + d_l + (\sum_{i=1}^{N-1} \left|d_i - \bar{d}\right|)}{(df + d_l + (N-1) \cdot \bar{d})} \tag{4}$$

where N is the number of solutions d_i. df and d_l are the Euclidean distances between the extreme solutions and the boundary solutions, and $\bar{d}$ is the average of all the consecutive distances d_i for $i = 1, 2, 3, \ldots, N-1$. Smaller value of Δ indicates a better and more uniform distribution of the solutions along the Pareto front.

4) NUMBER OF NON-DOMINATED SOLUTIONS

The number of non-dominated solutions measures the effectiveness of the optimization algorithm in terms of its ability to discover non-dominated solutions. NDS can be calculated using Equation 5.

$$NDS\left(N\right) = |P_s| \tag{5}$$

A higher NDS value suggests that the algorithm has found a larger set of optimal trade-offs between conflicting objectives, which is often the goal in the MOO problems.

5) GENERATIONAL DISTANCE (GD)

Generational distance metric, measures the distance between the obtained Pareto front (PS) by the proposed method and the true Pareto front PT. A lower GD value indicates better performance. GD can be calculated by the Equation 6.

$$GD\left(P_S, P_T\right) = \frac{\sqrt{\sum_{i=1}^{|P_S|} d_i^2}}{|P_S|} \tag{6}$$

where $|P_s|$ is the number of solutions in the Pareto set, P_T is the true Pareto front, d_i is the Euclidean distance between the solutions in P_s and the nearest solutions in P_T.

TABLE 2. Benchmarking mathematical functions.

Problem	n	Variable bounds	Objective functions	Optimal solutions	Shape
FON	3	$[-4,4]$	$f_1(x) = 1 - \exp\left(-\sum_{i=1}^{3}(x_i - \frac{1}{\sqrt{3}})^2\right)$ $f_2(x) = 1 - \exp\left(-\sum_{i=1}^{3}(x_i + \frac{1}{\sqrt{3}})^2\right)$	$x_1 = x_2 = x_3$	Non-Convex
KUR	3	$[-5,5]$	$f_1(x) = \sum_{i=1}^{n-1}\left(-10exp\left(-0.2\sqrt{x_i^2 + x_{i+1}^2}\right)\right)$ $f_2(x) = \sum_{i=1}^{n}(\lvert x_i\rvert^{0.8} + 5\sin(x_i^3))$	$x_i \in [-5,5]$ $i = 1,2,3,\ldots,n$	Non-convex disconnected
ZDT1	30	$[0,1]$	$f_1 = (x) = x_1$ $f_2(x) = g(x)\left[1 - \sqrt{\frac{x_1}{g(x)}}\right]$ Where $g(x) = 1 + 9(\sum_{i=2}^{n} x_i)/(n-1)$	$x_1 \in [0,1]$ $x_i = 0$ $i = 1,2,3,\ldots,n$	Convex
ZDT2	30	$[0,1]$	$f_1 = (x) = x_1$ $f_2(x) = g(x)\left[1 - (x_1/g(x))^2\right]$ Where $g(x) = 1 + 9(\sum_{i=2}^{n} x_i)/(n-1)$	$x_1 \in [0,1]$ $x_i = 0$ $i = 1,2,3,\ldots,n$	Non- convex
ZDT3	30	$[0,1]$	$f_1 = (x) = x_1$ $f_2(x) = g(x)\left[1 - \sqrt{\frac{x_1}{g(x)}} - \frac{x_1}{g(x)}\sin(10\pi x_1)\right]$ $g(x) = 1 + 9(\sum_{i=2}^{n} x_i)\Big/(n-1)$	$x_1 \in [0,1]$ $x_i = 0$ $i = 1,2,3,\ldots,n$	Convex, disconnected
ZDT6	10	$[0,1]$	$f_1(x) = 1 - \exp(-4x_1)\,sin^6(6\pi x_1)$ $f_2(x) = g(x)\left[1 - (\frac{f_{1(x)}}{g(x)})^2\right]$ $g(x) = 1 + 9\left[(\sum_{i=2}^{n} x_i)/n - 1\right]^{0.25}$	$x_1 \in [0,1]$ $x_i = 0$ $i = 1,2,3,\ldots,n$	non-uniform, non-convex

V. EXPERIMENTAL DESIGN AND RESULTS ANALYSIS

This section includes nine sub-sections: the first covers the steps of experimental design, six sub-sections present the results of the proposed algorithm in comparison to state-of-the-art benchmarks, and the final two sub-sections address complexity analysis and parameter sensitivity.

A. EXPERIMENTAL DESIGN

The experimental design for the proposed algorithm was evaluated across a range of six mathematical functions, each possessing diverse shapes and level of complexity, FON, KUR, ZDT1, ZDT2, ZDT3, and ZDT6. These functions were characterized by varying acceleration parameters c, as explained in Table 3. Additionally, the parameters for MEPSOLA, including the embedded Tabu search parameters, are outlined in the same table.

The proposed method compared with well-established benchmarking MOO algorithms, NSGA-II [44], NSGA-III [45], and MOEA [46]. These benchmark algorithms are recognized for their robust performance across a wide range of MOO problems and have been extensively validated in the literature. To ensure parity in our comparative analysis, the parameter settings for the benchmarking were directly aligned with those recommended in their original publications with minor tuning.

TABLE 3. The experimental design parameters.

MEPSOLA parameters	
Population size	100
Inertia weight (W)	1
Acceleration parameter (C)	0.2
Mutation ratio	0.2
Number of iterations	100
number of decision variables	3
Tabu search parameters	
Max neighbors	10
Max iterations	80
Tabu list size	10
Mathematical function parameters	
Mathematical function	Acceleration value
FON, ZDT1	0.2
ZDT2, ZDT6, ZDT3	0.4
KUR	0.7

The experimental simulations were implemented using the Python programming language in two platforms the

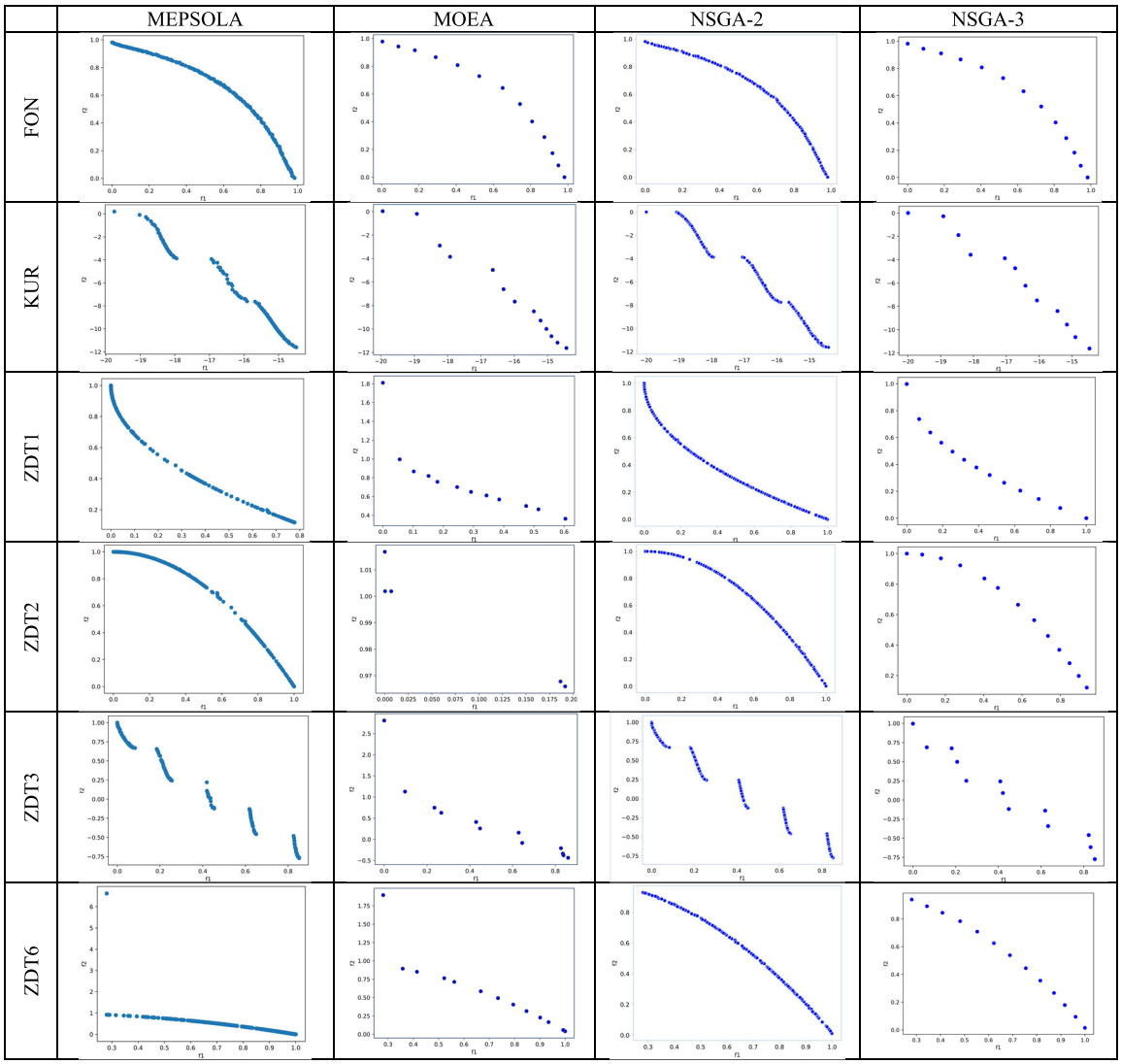

FIGURE 2. Visualizing a Pareto front obtained from the proposed method and the benchmarks MOEA, NSGA-2, and NSGA-3 based on mathematical functions FON, KUR, ZDT1, ZDT2, ZDT3, and ZDT6.

PYCharm Community Edition 2023.1.2 and Google Colab. The hardware setup utilized a 12th generation Intel Core i5 of 4.40 GHz processor with 10 cores and 16 GB of DDR4 RAM, ensuring that the computational environment was sufficiently capable of handling the intensive calculations required by these optimization algorithms.

B. RESULTS

In complex optimization scenarios, the efficiency of an optimization algorithm is crucial. This article introduces an enhanced version of MOPSO called MEPSOLA, specifically tailored for addressing complex optimization problems with high dimensionality. MEPSOLA integrates a range of sophisticated techniques to improve its performance.

The experimental results analysis presented in this article aims to demonstrate the algorithm's effectiveness in navigating the complexities of MOO. The results, illustrated in Fig. 2, display the Pareto front obtained from one run from ten experimental runs for MEPSOLA and each benchmark. This visual representation serves as evidence of the algorithm's capability, compared to other benchmarking algorithms, to produce a diverse and evenly distributed set of solutions that closely approximate the true Pareto front.

To assess performance across multiple runs, we utilize box plots as a graphical representation of the distribution of results. By employing box plots to visualize the outcomes of multiple runs of the MOO algorithm, we gain valuable insights into its behavior, variability, and overall performance across various executions.

C. FONSECA FUNCTION (FON) RESULTS

1) SET COVERAGE RESULTS

The SC metric results of the FON function, SC measure the level of dominance exhibited by one set of solutions

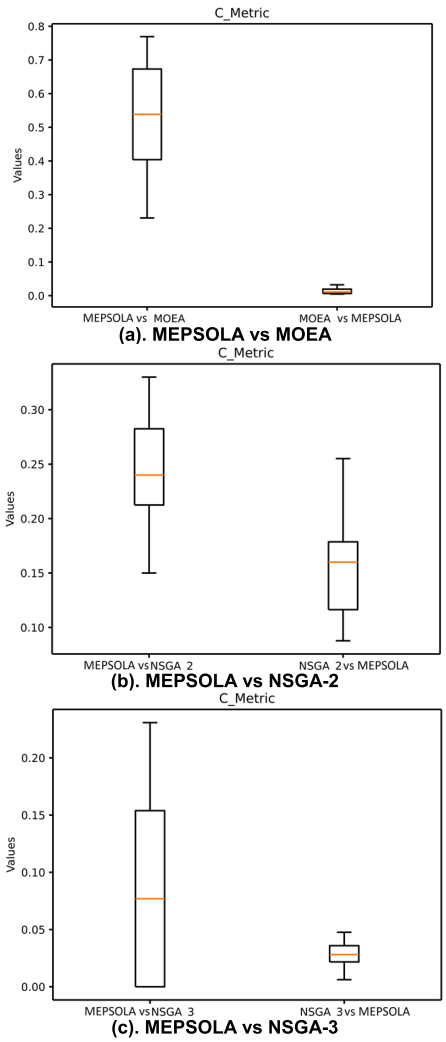

FIGURE 3. Set coverage measure results for FON function.

over another. In the comparison between MEPSOLA and MOEA as illustrated in Fig. 3 (a), MEPSOLA's solutions demonstrate significant dominance over those in MOEA, as evidenced by SC interquartile ranging from Q1 of 0.404 to Q3 of 0.673. The median SC value of 0.53 indicates that in the majority of runs, MEPSOLA's solutions dominate over MOEA's solutions. Conversely, MOEA exhibits minimal dominance over MEPSOLA's solutions, with SC interquartile ranging from Q1 of 0.006 to Q3 of 0.019, and a median SC value of 0.01, suggesting rare instances of dominance.

Similarly, when comparing MEPSOLA with NSGA-2 as depicted in Fig 3. (b), MEPSOLA showcases a notable level of dominance, with SC interquartile values ranging from 0.21 to 0.28 and a median SC value of 0.24. In contrast, NSGA-2 demonstrates less dominance over MEPSOLA, with SC interquartile values ranging from Q1 of 0.11 to Q3 of 0.17 and a median SC value of 0.16.

In MEPSOLA against NSGA-3 as illustrated in Fig. 3(c), MEPSOLA's solutions exhibit a median SC value of 0.07, indicating lower dominance compared to MOEA and NSGA-2. Conversely, NSGA-3's solutions display a less degree of dominance over MEPSOLA's, with a median SC value of 0.02, which falls between MOEA's and NSGA-2's levels of dominance.

2) HYPER VOLUME RESULTS

The HV metric results, illustrated in Fig. 4 (a), represent the degree of coverage of the generated Pareto front in relation to the objective space. MEPSOLA achieves the HV interquartile values, ranging from Q1 of 567.4 to Q3 of 680.6, indicating extensive coverage of the objective space. The median HV value of 655.11 suggests that MEPSOLA consistently produces Pareto fronts that capture a considerable region of the optimal set across several runs.

In contrast, MOEA exhibits substantially lower HV values, indicating limited coverage of the objective space. The maximum HV value of 24.67 is considerably lower to MEPSOLA's, and the median HV of 24.61, along with narrow interquartile ranges from Q1 of 24.59 to Q3 of 24.64, suggests that MOEA's Pareto fronts fall short of extending across the optimal set. Subsequent, the NSGA-2 demonstrates consistent HV values within a narrow interquartile range, from Q1 of 188.12 to Q3 of 188.12. The median HV of 188.21 indicates a moderate coverage of the objective space that remains consistent across runs but is not as expansive as MEPSOLA's.

Similarly, NSGA-3 exhibits consistent but low HV values, the interquartile ranging from Q1 of 24.69 to Q3 of 24.71. The median HV of 24.7 and interquartile values suggest performance similar to MOEA in terms of HV, with a Pareto front covering only a minimal portion of the optimal set. In the case of the FON function, MEPSOLA outperforms MOEA, NSGA-2, and NSGA-3 significantly in terms of HV.

3) DELTA MEASURE RESULTS

The delta measure results for the FON function, as shown in Fig. 4 (b), measure the proximity between the true Pareto front and the approximated front produced by the algorithm. In the case of MEPSOLA, the range of delta interquartile values spans from Q1 of 0.801 to Q3 of 0.868, with a median of 0.826. While the maximum value is not optimal, the relatively lower minimum and median values suggest that MEPSOLA maintains a satisfactory distribution of solutions.

For MOEA, the delta interquartile values vary from Q1 of 0.677 to Q3 of 0.679, with a median of 0.679. These values indicate a tight spread, also they may not cover the Pareto-optimal region as extensively as desired. Similarly, for NSGA-2, the delta interquartile values range from Q1 of 0.674 to Q3 of 0.683, with a median of 0.678. These findings imply a consistent but moderate uniformity without exceptional coverage. In the case of NSGA-3, the median delta value is 0.675, closely resembling those of MOEA and NSGA-2. This suggests a consistent yet potentially narrower spread, and could also indicates lacking the diversity ideally desired [47].

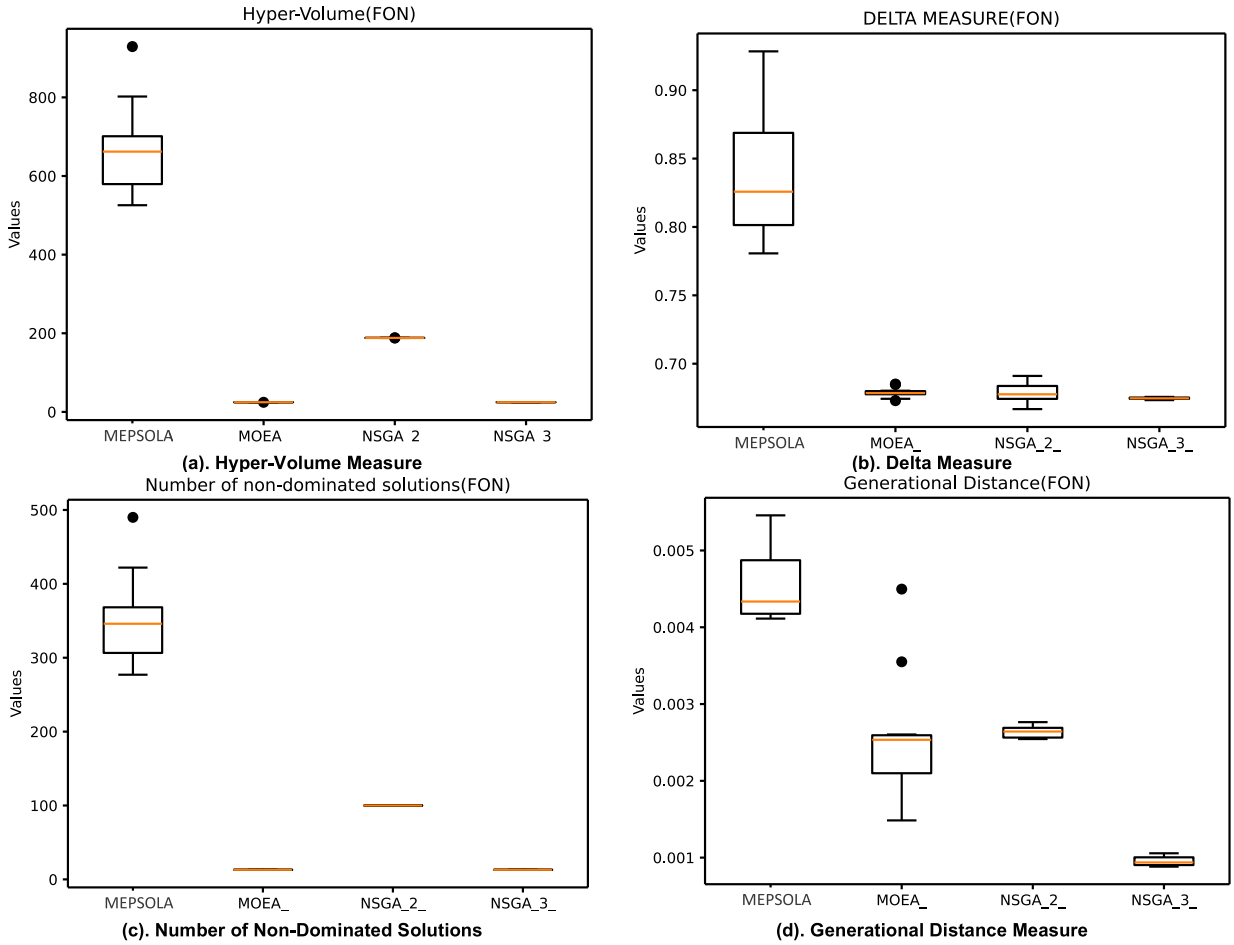

FIGURE 4. FON function results.

4) NUMBER OF NON-DOMINATED SOLUTIONS RESULTS

The NDS metric results depicted in Fig. 4 (c) for the FON function, the measure evaluates the efficacy of optimization algorithms in discovering a varied array of Pareto-optimal solutions. In the case of MEPSOLA, the findings reveal a robust efficacy to identify a considerable number of non-dominated solutions, ranging from a minimum count of 277 to a maximum of 422. With a median count of 341, MEPSOLA consistently discovers a significant number of optimal solutions across several runs.

Conversely, MOEA exhibits lower NDS values, all fixed at 13. This consistent figure implies that MOEA may have a narrower scope in exploring the solution space. Furthermore, the NSGA-2 yields a consistent count of 100 non-dominated solutions across all statistical measures, surpassing MOEA's count but still limiting the variety and quantity of solutions compared to MEPSOLA. However, NSGA-3 yields result similar to MOEA, with a fixed count of 13 non-dominated solutions. This similarity indicates a comparable limitation in the diversity and range of solution space exploration. In the context of the FON function, MEPSOLA outperforms MOEA, NSGA-2, and NSGA-3 in terms of the NDS metric, showcasing its ability to generate a vast and varied set of Pareto-optimal solutions.

5) GENERATIONAL DISTANCE RESULTS

The GD measure evaluates the proximity of generated solutions to the true Pareto front. In Fig. 4 (d), MEPSOLA obtained GD interquartile values ranging from Q1 of 0.0041 to Q3 of 0.0048, with a median of 0.0043, indicating consistently close approximation to the true Pareto front. While in MOEA GD interquartile values ranging from 0.002 to 0.0025 and a median of 0.0023, suggesting less consistent approximation. Furthermore, in NSGA-2 presents a narrower range of GD interquartile values from Q1 of 0.0025 to Q3 of 0.0026 with a median of 0.0026, indicating relatively consistent performance and slightly less compared to MOEA.

The NSGA-3 results demonstrate the closest approximation to the Pareto front with a median GD value of 0.0009, reflecting tight convergence across runs. The range of interquartile values from Q1 of 0.0009 to Q3 of 0.001 further confirm this consistent performance. Overall, NSGA-3 exhibits the closest convergence to the Pareto front,

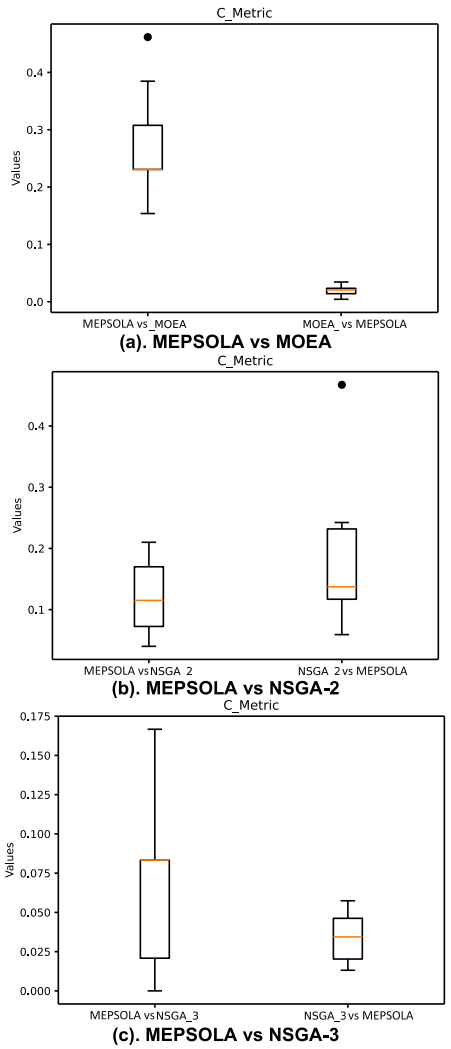

FIGURE 5. Set coverage measure results for KUR function.

followed by MOEA, while MEPSOLA and NSGA-2 also perform well, albeit with slightly wider ranges suggesting marginally less tight convergence compared to NSGA-3.

D. KURSAWE FUNCTION (KUR) RESULTS

1) SET COVERAGE RESULTS

The analysis of the SC results for the KUR function, as illustrated in Fig. 5 (a), shows that MEPSOLA significantly dominates the solutions provided by MOEA. The SC interquartile values range from a Q1 of 0.23 to Q3 of 0.31, with a median of 0.23, indicating that a large number of MEPSOLA's solutions outperform MOEA's. Conversely, MOEA exhibits minimal dominance over MEPSOLA, with interquartile values ranging from Q1 of 0.014 to Q3 of 0.023 and a median of 0.02, suggesting rare occurrences where MOEA's solutions surpass those of MEPSOLA.

In comparison MEPSOLA against NSGA-2 in the Fig. 5(b), MEPSOLA displays some dominance, with SC values extending from 0.04 to 0.21. A median SC of 0.12 and an interquartile range from Q1 of 0.07 to Q3 of 0.17 indicate that MEPSOLA's solutions frequently cover those from NSGA-2, albeit to a moderate extent. However, NSGA-2 shows some dominance over MEPSOLA, with an interquartile SC value of Q1 of 0.116 and Q3 of 0.227, though with a moderate median of 0.13. Furthermore, the comparison of MEPSOLA against NSGA-3 as shown in the Fig. 5 (c), MEPSOLA's solutions exhibit a dominance over NSGA-3 median SC value of 0.08, slightly less dominance compared to MOEA and NSGA-2. NSGA-3 demonstrates limited dominance over MEPSOLA with a median SC of 0.03.

These SC outcomes for the KUR function reveal that MEPSOLA's solutions significantly outperform MOEA and moderately surpass NSGA-3, while NSGA-2 exhibits only minimal dominance over MEPSOLA.

2) HYPER VOLUME RESULTS

The results of the HV metric for the KUR function as shown in Fig. 6 (a), highlight MEPSOLA exceptional performance, presenting HV values ranging from a minimum of 5391.07 to a maximum of 18171.68. This range underscores MEPSOLA's capability to extensively cover the objective space. The median HV of 9397.94 indicates that MEPSOLA consistently produces solutions encompassing a significant area of the space, confirmed by the interquartile range from Q1 of 8693.74 to Q3 of 12604.38, demonstrating the algorithm's robust performance. In contrast, MOEA exhibits considerably lower HV values, peaking at just 225.33, which signals a very limited exploration of the objective space. The median HV value of 217.76, coupled with a narrow interquartile range extending from a Q1 of 217.43 and Q3 of 218.05, illustrates MOEA's limited capability in covering the space as MEPSOLA's solutions do.

The NSGA-2 result presents moderate HV values, with ranging from 1709.9 to 1743.53. The median HV of 1723.68 and a relatively narrow interquartile range from Q1 of 1718.39 to Q3 of 1729.16 indicate that NSGA-2 covers a greater portion of the objective space than MOEA, yet it does not reach the expansive coverage achieved by MEPSOLA. Similarly, NSGA-3's performance aligns more closely with MOEA's, displaying HV values from 194.57 to 198.73. The median HV of 196.68 and interquartile values from Q1 of 196.32 to Q3 of 197.02 suggest that NSGA-3's Pareto fronts are also confined in scope, covering only a minimal section of the objective space. In the analysis of the KUR function, MEPSOLA markedly surpasses MOEA, NSGA-2, and NSGA-3 in terms of hyper-volume. This demonstrates MEPSOLA's superior ability to generate solutions that not only closely approach the true Pareto front but also offer a comprehensive and diverse representation of optimal solutions.

3) DELTA MEASURE RESULTS

In the Delta Measure results for the KUR function, depicted in Fig. 6 (b), MEPSOLA demonstrates a notable spread in its solution distribution, albeit with slight non-uniformity. The

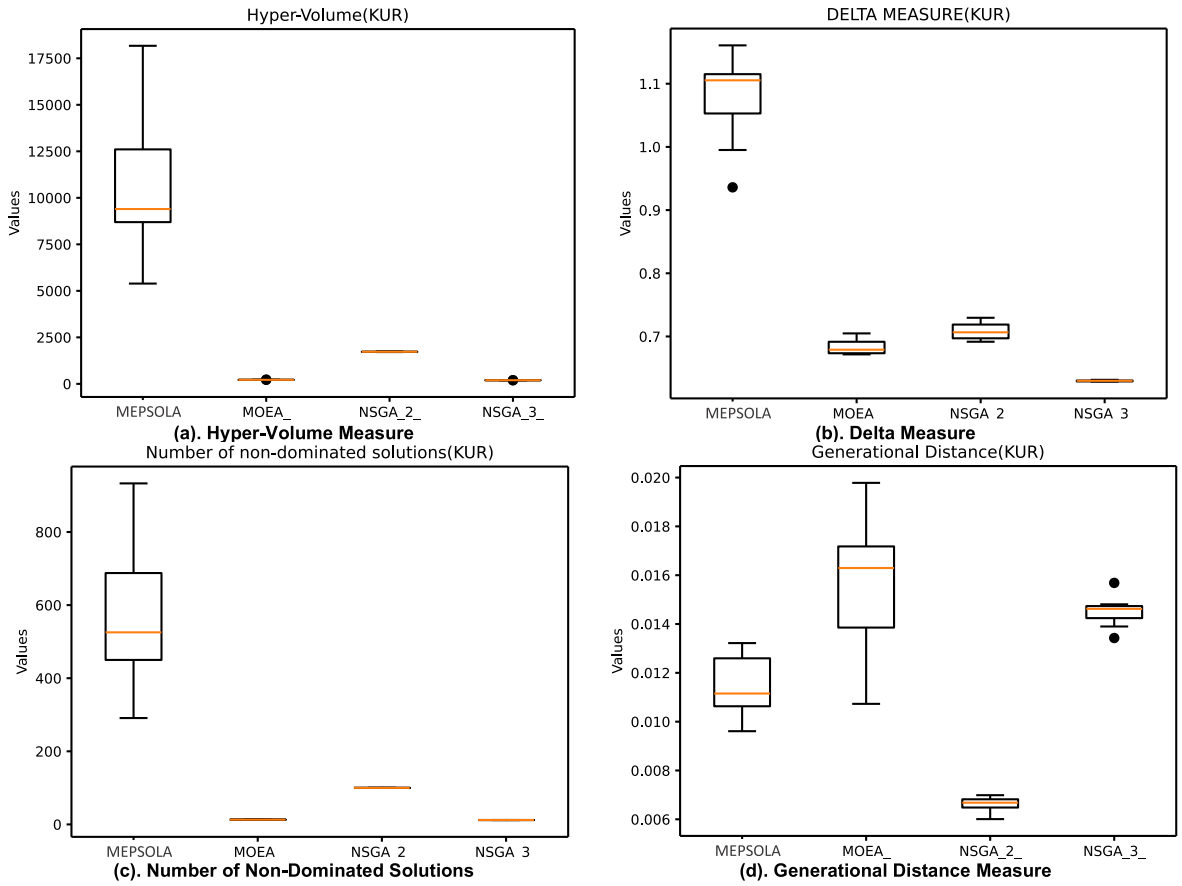

FIGURE 6. KUR function results.

median value of 1.107 suggests a generally broad and diverse distribution. The interquartile range for Q1 of 1.076 and Q3 of 1.116 also indicates some non-uniformity but may signify proficient exploration of the search space, which could be interpreted alongside other measures. In contrast, MOEA's Delta values show a narrower scope, with an interquartile range from Q1 of 0.673 to Q3 of 0.691, and a median of 0.679. NSGA-2 exhibits a fair distribution, with a Delta median value of 0.706. The interquartile range for Q1of 0.697 and Q3 of 0.719 suggests reasonable uniformity in solution spread. On the other hand, NSGA-3 reveals the most narrowly focused distribution among the compared algorithms, with a median of 0.629. The quartile values for Q1 of 0.628 and Q3 of 0.63 indicate a highly concentrated distribution.

4) NUMBER OF NON-DOMINATED SOLUTIONS

In the NDS measure for the KUR function, MEPSOLA demonstrates a robust ability to discover a wide array of non-dominated solutions, as illustrated in Fig. 6 (c). The results show a median count of 525.5 non-dominated solutions, with an interquartile range stretching from Q1 at 450 to Q3 at 687.7, highlighting MEPSOLA's consistent effectiveness in identifying numerous optimal solutions. In contrast, MOEA identifies only a small number of non-dominated solutions, consistently capped at 13 across all statistical measures. NSGA-2 maintains a steady count of 100 non-dominated solutions across all statistical measures, indicating a moderate exploration capability. Similarly, NSGA-3 mirrors MOEA with a limited count of 12 non-dominated solutions across all statistical measures, suggesting a constrained exploration of optimal solutions. In the results for the KUR function, MEPSOLA markedly surpasses MOEA, NSGA-2, and NSGA-3 in identifying a broad spectrum of non-dominated solutions.

5) GENERATIONAL DISTANCE RESULTS

In the GD Measure results for the KUR function, as shown in Fig. 6 (d), MEPSOLA shows a spectrum of GD values, with the interquartile ranging from 0.0106 for Q1 to 0.0125 for Q3, centering around a median value of 0.0111. This range indicates that MEPSOLA's solutions generally lie close to the Pareto front.

In comparison, MOEA exhibits a broader range of GD values, spanning from Q1 of 0.0138 to Q3 of 0.0171. Notably, the upper limit of this range is the highest among all the algorithms. The median GD value for MOEA, at 0.0163, suggests a slightly less consistent approximation to the Pareto front compared to MEPSOLA. The NSGA-2 presents the

lowest GD values, indicating a closer approximation to the Pareto front than the others. The interquartile range from a Q1 of 0.0064 to a Q3 of 0.0068, with the median at 0.0066, reflects NSGA-2's effectiveness in this regard. On the other hand, NSGA-3's GD values are generally higher than those of NSGA-2. The median value of 0.0146, along with quartile values of 0.0142 for Q1 and 0.0147 for Q3, suggests that NSGA-3's solutions are somewhat further from the Pareto front compared to NSGA-2.

E. ZDT1 FUNCTION RESULTS

1) SET COVERAGE RESULTS

The SC measure results for the ZDT1 function, in Fig. 7 (a), the result highlight MEPSOLA's significant dominance over MOEA. The interquartile range spans from Q1 at 0.923 to Q3 and median values reaching 1, indicating that almost all of MEPSOLA's solutions dominate those of MOEA. In contrast, MOEA exhibits minimal dominance over MEPSOLA, with most SC values at 0 and a maximum of only 0.0116, reflecting almost negligible coverage over MEPSOLA's solutions.

Moving on to the comparison between MEPSOLA and NSGA-2, as depicted in Fig. 7 (b), MEPSOLA again demonstrates dominance, with a median SC of 0.73. The interquartile range from Q1 of 0.7025 to Q3 of 0.765 indicates that a significant majority of MEPSOLA's solutions are superior to those of NSGA-2. However, NSGA-2 does exhibit less dominance over MEPSOLA, albeit less obviously, with an interquartile range from Q1 of 0.1726 to Q3 of 0.2154 and a median of 0.1913.

In the results of MEPSOLA against NSGA-3, as shown in Fig. 7 (c), MEPSOLA maintains strong dominance, with identical median and quartile SC values at 0.7692, demonstrating consistent and substantial coverage. In contrast, NSGA-3 shows the lowest level of dominance over MEPSOLA, with a median SC of 0.0349. These SC results for the ZDT1 function underscore MEPSOLA's impressive capability in consistently outperforming MOEA, NSGA-2, and NSGA-3.

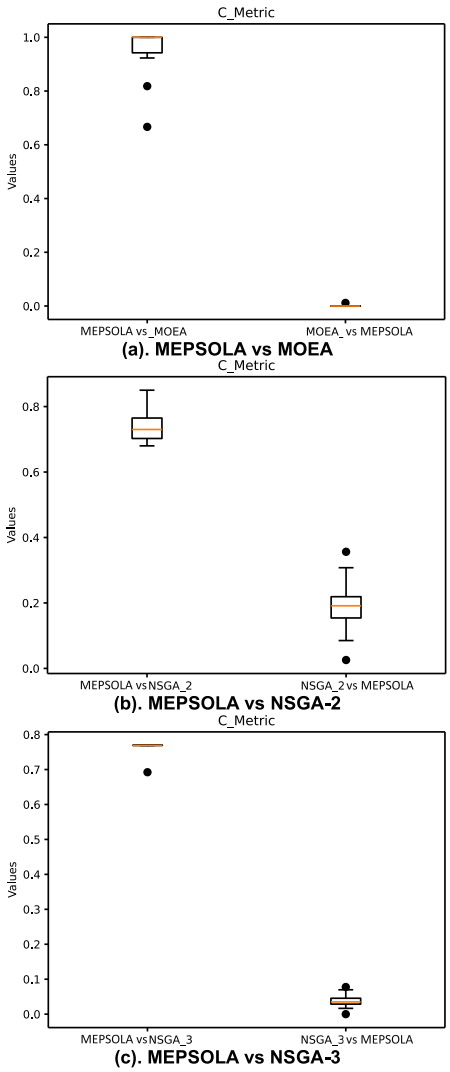

FIGURE 7. Set coverage measure results for ZDT1 function.

2) HYPER VOLUME RESULTS

The HV results for the ZDT1 function in Fig. 8 (a), reveals MEPSOLA's notable performance, with HV values showcasing a median HV of 261.54. The interquartile range, stretching from Q1 of 203.97 to Q3 of 367.28, indicates MEPSOLA consistently generates Pareto fronts that cover a substantial portion of the optimal set across different runs. In contrast, MOEA exhibits limited HV performance, with its highest HV value only reaching 60.51 and a median HV of 32.98. The quartile values, ranging from Q1 at 31.66 to Q3 at 40.16, suggest MOEA's Pareto fronts are significantly less effective in covering the objective space compared to MEPSOLA.

For NSGA-2, the HV values are more consistent than MOEA, but they still remain lower than those of MEPSOLA. With a median HV of 241.663, NSGA-2 indicates a moderate coverage of the objective space, with quartile values at Q1 of 241.24 and Q3 of 242.39. Conversely, NSGA-3 exhibits the lowest HV value, with a median HV of 31.55, along with quartile values of Q1 at 31.53 and Q3 at 31.56, indicating NSGA-3's Pareto fronts are the least effective in spanning the objective space. These HV results for the ZDT1 function underscore MEPSOLA's superior capability in generating Pareto fronts that not only approach the true Pareto front but also span a broader and more significant portion of the optimal set.

3) DELTA MEASURE RESULTS

The results of the delta measure for the ZDT1 function, shown in Fig. 8 (b), underscore MEPSOLA's performance in evenly distributing solutions along the Pareto front. With a median value of 1.025 and quartile values of Q1 at 0.988 and Q3 at 1.049, MEPSOLA showcases a broad yet slightly varied distribution along the Pareto front. In comparison, MOEA

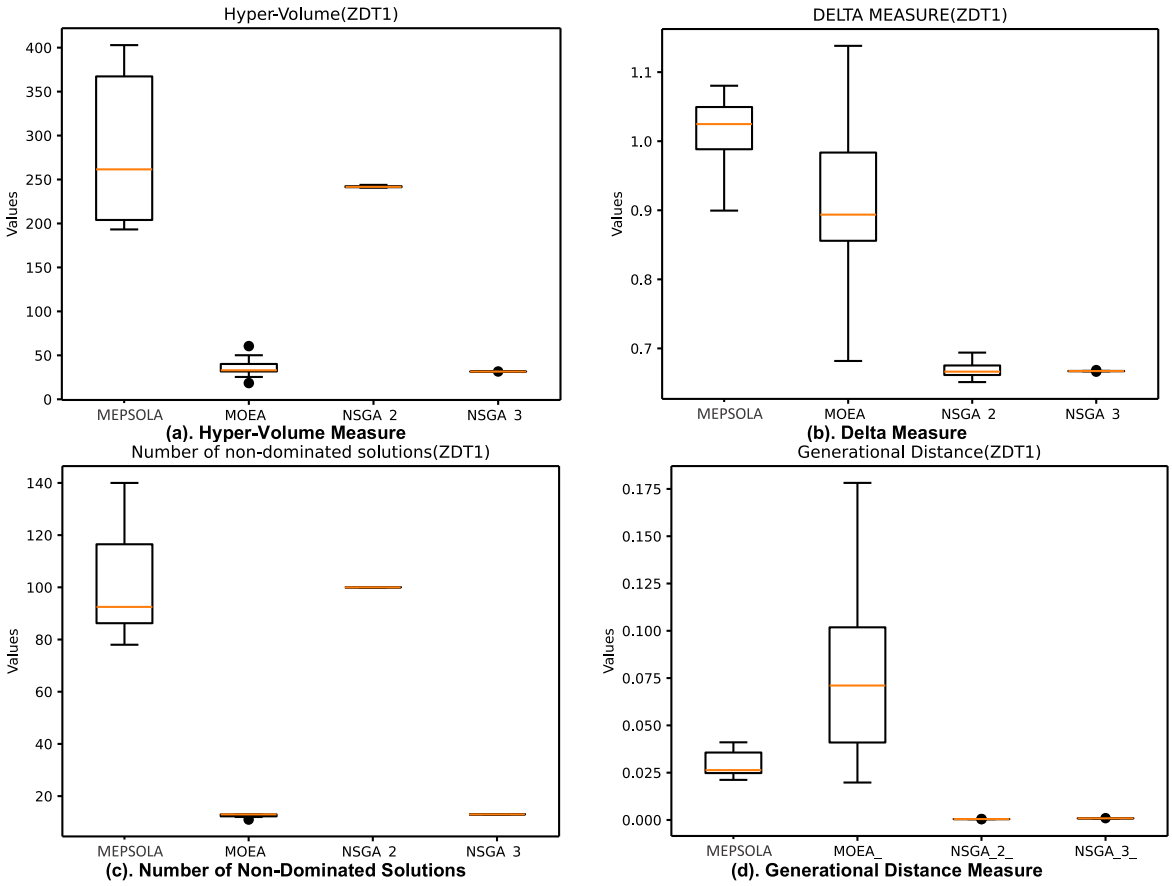

FIGURE 8. ZDT1 function results.

exhibits a slightly lower delta value than MEPSOLA, with a median value of 0.8936. The quartile range spans from Q1 of 0.8558 to Q3 of 0.9835.

For NSGA-2 and NSGA-3, both algorithms demonstrate lower Delta values. NSGA-2's Delta values range from 0.6512 to 0.6939, with a median of 0.6663. Similarly, NSGA-3 exhibits Delta quartile values from Q1 of 0.6669 to Q3 of 0.6673, with a median of 0.6671. Despite the slightly higher Delta values, MEPSOLA's overall performance in the ZDT1 function remains robust, evidenced by its strong outcomes in SC, HV, and the NDS metrics.

4) NUMBER OF NON-DOMINATED SOLUTIONS

The NDS measure results for the ZDT1 function, shown in Fig. 8 (c), highlight MEPSOLA's strong performance, with a median of 92.5, and the interquartile range, spanning from Q1 at 86.25 to Q3 at 116.5, underscores MEPSOLA's consistent ability to identify a high number of quality solutions. In contrast, MOEA exhibits significantly fewer non-dominated solutions, with all statistical values notably lower than MEPSOLA's, ranging from a minimum of 11 to a maximum of 13. The median and quartile values consistently hover around 13, indicating that MOEA struggles to find a diverse set of optimal solutions.

The NSGA-2 maintains a consistent count of 100 non-dominated solutions across all statistics, surpassing MOEA's performance but falling short of MEPSOLA's maximum value.

Similarly, NSGA-3 also demonstrates a limited range of NDS, akin to MOEA, with 13 NDS across all statistics, indicating a constrained exploration of the solution space. In the context of the ZDT1 function, MEPSOLA outperforms MOEA, NSGA-2, and NSGA-3 in terms of the identified NDS, showcasing its effectiveness in exploring the search space and discovering a greater number of high-quality solutions.

5) GENERATIONAL DISTANCE RESULTS

The GD results for the ZDT1 function, illustrated in Fig. 8 (d), provide intriguing insights into the performance of MEPSOLA compared to other algorithms. MEPSOLA obtained a median GD value of 0.0263, with quartile values of 0.0247 for Q1 and 0.0356 for Q3.

In contrast, MOEA displays a wider variation in GD values, with a median GD value of 0.0711, notably higher than that of MEPSOLA. The interquartile range spans from 0.0409 for Q1 to 0.1018 for Q3, indicating inconsistencies

TABLE 4. The set coverage measure results for ZDT2, ZDT3 and ZDT6.

Function		MEPSOLA vs MOEA	MOEA vs MEPSOLA	MEPSOLA vs NSGA-2	NSGA-2 vs MEPSOLA	MEPSOLA vs NSGA-3	NSGA-3 vs MEPSOLA
ZDT2	Min	**0.83333**	0	**0.06**	0.00166	**0.07692**	0
	Max	**1**	0	**0.16**	0.06517	**0.23077**	0.00875
	Median	**0.96154**	0	**0.11**	0.01059	**0.15385**	0.00152
	Q1	**0.91259**	0	**0.07**	0.00331	**0.07692**	0
	Q3	**1**	0	**0.12**	0.03632	**0.15385**	0.00449
ZDT3	Min	**1**	0	**0.03**	0.02732	0	**0.00182**
	Max	**1**	0	0.09	**0.1532**	**0.38462**	0.03409
	Median	**1**	0	0.07	**0.09011**	**0.19872**	0.0116
	Q1	**1**	0	0.0525	**0.05986**	**0.09615**	0.00443
	Q3	**1**	0	0.0875	**0.11092**	**0.25**	0.01666
ZDT6	Min	**0.84615**	0	**0.58**	0.0012	**0.53846**	0.0012
	Max	**1**	0.00777	**0.82**	0.0462	**0.92308**	0.0462
	Median	**0.88462**	0.00454	**0.69**	0.01149	**0.65385**	0.01149
	Q1	**0.84615**	0.00152	**0.6425**	0.00583	**0.61538**	0.00631
	Q3	**0.92308**	0.00619	**0.7225**	0.01331	**0.84615**	0.01331

in MOEA's performance, with certain runs deviating notably from the Pareto front.

Conversely, NSGA-2 and NSGA-3 demonstrate low GD values, signifying a more consistent proximity to the Pareto front. NSGA-2 boasts a median GD of just 0.0004, while NSGA-3's median GD is 0.0008, both substantially lower than those of MEPSOLA and MOEA.

F. ZDT2, ZDT3, AND ZDT6 FUNCTIONS RESULTS

1) SET COVERAGE RESULTS

The remaining results for ZDT2, ZDT3, and ZDT6 functions, as illustrated in Table 4, starting with SC measure, the results highlight the robust performance of MEPSOLA in dominating the solutions of other competitors, MOEA, NSGA-2, and NSGA-3. For the ZDT2 function, MEPSOLA demonstrates significant dominance over MOEA, as indicated by the median SC value reaching 0.96154, whereas MOEA exhibits no dominance over MEPSOLA. This dominance is also observed, albeit to a lesser extent, against NSGA-2 and NSGA-3, with median SC values of 0.11 and 0.15385, respectively, whereas the methods obtained lower SC values with 0.01059 and 0.00152, respectively, suggesting that a considerable proportion of MEPSOLA's solutions outperform those from NSGA-2 and NSGA-3.

In the ZDT3 function, MEPSOLA's superiority continues to excel, particularly evident in its total dominance over MOEA, where both the median and maximum SC values peak at 1. MOEA exhibits no dominance over MEPSOLA. While against NSGA-2, it shows some advantage compared to MEPSOLA with a median of 0.09011, whereas MEPSOLA obtained a 0.07 SC value. However, its performance against NSGA-3 indicates significant but not absolute dominance, as reflected by a median SC value of 0.19872, compared to NSGA-3's value of 0.0116.

In the context of the ZDT6 function, MEPSOLA's strength is further confirmed, showing strong dominance over MOEA. This is underscored by SC values with a minimum of 0.84615 and a median of 0.88462, compared to MOEA's median value of 0.00454. Against NSGA-2, MEPSOLA's dominance is marked, with a median SC value of 0.69, whereas NSGA-2 only achieved 0.01149 median. Additionally, MEPSOLA maintains a substantial edge over NSGA-3, as indicated by the median SC value of 0.65385, compared to NSGA-3's value of 0.01149.

These findings collectively present MEPSOLA as a effective algorithm in MOO, consistently outperforming MOEA, NSGA-2, and NSGA-3 across various test functions. The consistent and significant dominance of MEPSOLA in the Set Coverage metric across majority of the three functions not only underscores its efficacy in producing competitive solution sets but also cements its position as a preferred choice for tackling complex optimization challenges in the multi-objective domain.

2) HV, DELTA, NDS, AND GD MEASURES RESULTS

The remaining results for the HV, delta, NDS, and GD measures for the mathematical functions ZDT2, ZDT3, and ZDT6 are illustrated in Table 5.

In the HV metric MEPSOLA exhibits notable superiority, showcasing its robust capacity to cover an extensive area of the objective space. For example, on the ZDT2 function, MEPSOLA achieves a median HV value of 957.5, surpassing MOEA's median of 15.2, NSGA-2's median of 192.5, and

TABLE 5. The results for four metrics HV, Δ, NDS and GD for three mathematical functions ZDT2, ZDT3 and ZDT6.

		ZDT2				ZDT3				ZDT6			
Function		MEPSOLA	MOEA	NSGA2	NSGA3	MEPSOLA	MOEA	NSGA2	NSGA3	MEPSOLA	MOEA	NSGA2	NSGA3
HV	Min	**693.6**	13	192.3	24	**809.5**	20.8	255.8	26.7	**1519.5**	31.1	181.8	23.6
	Max	**1365.4**	19.7	192.7	25.2	**1378**	64	258.6	33.4	**8355.5**	119.1	182.5	24.9
	Median	**957.5**	15.2	192.5	25.2	**1113.2**	33.2	257.3	33.4	**2862.7**	52.4	181.9	23.7
	Q1	**803.0**	14.1	192.3	25.2	**1043.8**	25	256.6	30.9	**2471.9**	40.6	181.8	23.7
	Q3	**1042.9**	16	192.6	25.2	**1317.1**	40	257.8	33.4	**5418.5**	59.9	181.9	23.8
Δ	Min	0.98091	0.82796	0.65155	0.64612	1.06796	0.78464	0.76266	0.61386	1.22732	0.74465	0.62248	0.5767
	Max	1.18929	1.41667	0.68341	0.64845	1.15998	0.91545	0.78131	0.63613	1.38534	1.25628	0.64379	0.60211
	Median	1.07969	1.08146	0.66363	0.64628	1.12535	0.84025	0.77271	0.63561	1.30178	1.04605	0.63298	0.59869
	Q1	1.04344	1.02912	0.65633	0.64617	1.10819	0.80069	0.76841	0.63439	1.26747	0.9306	0.62924	0.59727
	Q3	1.10933	1.23568	0.67383	0.64638	1.13852	0.87755	0.77821	0.63659	1.32508	1.10321	0.64121	0.59974
NDS	Min	**394**	11	100	13	**359**	11	100	12	**298**	13	100	13
	Max	**706**	13	100	13	**586**	13	100	13	**895**	13	100	13
	Median	**513**	13	100	13	**486**	12	100	13	**442**	13	100	13
	Q1	**459**	13	100	13	**443**	12	100	12	**316**	13	100	13
	Q3	**588**	13	100	13	**541**	13	100	13	**682**	13	100	13
GD	Min	0.00114	0.00221	0.00048	0.00056	0.00105	0.01586	0.00069	0.00186	0.0346	0.04233	0.00055	0.00205
	Max	0.00203	0.00779	0.00053	0.00111	0.00194	0.15563	0.00077	0.01653	0.72636	0.2626	0.00079	0.00727
	Median	0.0017	0.00444	0.0005	0.00093	0.00123	0.05274	0.00073	0.00354	0.05502	0.13572	0.00067	0.00285
	Q1	0.00143	0.00339	0.00049	0.0009	0.00122	0.02227	0.00072	0.0034	0.04203	0.0861	0.00063	0.00265
	Q3	0.0019	0.00502	0.00052	0.00095	0.00131	0.10134	0.00075	0.00378	0.06081	0.16893	0.00073	0.00322

NSGA-3's median of 25.2. This highlights the algorithm's effectiveness in approximating the true Pareto front. Similarly, on ZDT3, MEPSOLA outperforms others with a median of 1113.2, compared to MOEA's median of 33.2, NSGA-2's median of 257.3, and NSGA-3's median of 33.4. Additionally, on ZDT6, MEPSOLA achieves the highest median of 2862.7, demonstrating its superior performance compared to other methods.

The Delta Measure results for ZDT2 and ZDT6 position MEPSOLA in the third rank, surpassing MOEA. The results exhibit a broader distribution of solutions across the Pareto front compared to NSGA-2 and NSGA-3, which obtain better uniformly distributed solutions. For ZDT2, MEPSOLA achieves a median value of 1.07969, and for ZDT6, it achieves 1.30178. However, MEPSOLA ranks last in ZDT3, with a median value of 1.12535.

In terms of NDS, MEPSOLA consistently outperformed its counterparts by discovering a larger quantity of optimal solutions, indicative of a diverse Pareto front. Specifically, within the ZDT2 function, MEPSOLA achieved a median NDS of 513, a substantial increase compared to MOEA's 13, NSGA-2's 100, and NSGA-3's 13. Similarly, in ZDT3, MEPSOLA obtained a median of 486, followed by NSGA-2 with 100, NSGA-3 with 13, and lastly, MOEA with a median value of 12. In ZDT6, MEPSOLA achieved 442, outperforming the other methods.

In the GD metric, MEPSOLA achieved commendable results, ranking third in ZDT2, ZDT3, and ZDT6, with median values of 0.0017, 0.00123, and 0.05502, respectively. These results demonstrate MEPSOLA's ability to maintain closeness to the Pareto front, indicating near-optimal convergence.

TABLE 6. Overall average median of the measures across all function.

Set Coverage Measure		
MEPSOLA vs MOEA	MEPSOLA vs NSGA-2	MEPSOLA vs NSGA-3
0.7692	**0.3258**	**0.3226**
MOEA vs MEPSOLA	NSGA-2 vs MEPSOLA	NSGA-3 vs MEPSOLA
0.006	0.101	0.0207

Measure	MEPSOLA	MOEA [46]	NSGA-2 [44]	NSGA-3 [45]
HV	**2541.32**	167.7	542.01	160.9
NDS	**413.33**	12.83	100	12.83
Δ	1.0771	0.8698	0.6869	0.6420
GD	0.0166	0.0471	0.0019	0.0039

G. OVERALL RESULTS

In the overall results of measures across all functions as shown in the Table 6, MEPSOLA demonstrates significant superiority across various metrics compared to its competitors. Notably, in Set Coverage measure, MEPSOLA

TABLE 7. Solutions initialization complexity calculation.

Technique	Complexity
Range Sampling	$O(n)$
Cartesian Product	$O(m^n)$
Random Sampling	$O(pop\_size)$
Particle Initialization	$O(pop\_size \cdot n)$
Cost Calculation	$O(pop\_size \cdot C)$
Non-Dominated Particles Identification	$O(pop\_size^2)$
Exemplar Assignment	$O(pop\_size \cdot E)$

TABLE 8. Exemplar selection complexity calculation.

Technique	Complexity
Initialization	$O(1)$
Exemplar Selection	$O(1)$
Tabu Search Execution	$O(T \cdot N)$

TABLE 9. Tabu search for exemplar selection complexity calculation.

Technique	Complexity
Neighborhood Generation	$O(N)$
Best Candidate Selection	$O(N)$
Updating Tabu List	$O(1)$
Loop Execution	$O(T \cdot N)$

TABLE 10. Overall MEPSOLA complexity calculation.

Technique	Complexity
Initialization	$O(n + m^n + pop\_size \cdot n + pop\_size \cdot C + pop\_size^2 + pop\_size \cdot E)$
Exemplar Check and Adjustment	$O(pop\_size)$
Velocity and Position Update	$O(pop\_size \cdot n)$
Cost Calculation and Mutation	$O(pop\_size \cdot (C + M))$
Exemplar Update	$O(pop\_size \cdot E)$
Repository Update	$O(pop\_size^2)$
Pareto Front Formation	$O(pop\_size^2)$

achieves an average median of 0.7692, while MOEA lags far behind at 0.006. Similarly, MEPSOLA outperforms NSGA-2 and NSGA-3, with average median set coverage measures of 0.3258 and 0.3226, respectively, compared to NSGA-2's 0.101 and NSGA-3's 0.0207. SC results confirmed that MEPSOLA has effectively explored and represented a broader range of optimal solutions across multiple objectives. In the HV metric, MEPSOLA's remarkable achievement with an average median of 2541.32 surpassing each of in second rank NSGA-2 of 542.01 followed by MOEA of 167.7 last NSGA-3 of 160.9 HV results highlights its superiority in covering a substantial portion of the Pareto front compared to its counterparts. The NDS metric results further underscore MEPSOLA's dominance, with an average median value of 413.33. In contrast, MOEA, NSGA-2, and NSGA-3 exhibit significantly lower values of 12.83, 100, and 12.83, respectively, emphasizing MEPSOLA's superior performance in identifying non-dominated solutions. Furthermore, in terms of the Delta measure and GD metrics, MEPSOLA maintains efficiency with modest median values of 1.0771 for the Delta

TABLE 11. Parameters sensitivity for MEPSOLA in ZDT-1 function.

Scenario No	W	c	PoP	Iteration
Original scenario	1	0.2	100	100
Scenario 1	2	0.2	100	100
Scenario 2	0.5	0.2	100	100
Scenario 3	1	0.01	100	100
Scenario 4	1	1	100	100
Scenario 5	1	0.2	100	30
Scenario 6	1	0.2	100	200
Scenario 7	1	0.2	50	100
Scenario 8	1	0.2	300	100

TABLE 12. Parameters sensitivity for MEPSOLA in FON function.

Scenario No	W	c	PoP	Iteration
Original scenario	1	0.2	100	100
Scenario 1	1	0.9	200	100
Scenario 2	1	0.9	50	100
Scenario 3	1	0.9	100	100
Scenario 4	0.7	0.9	200	100
Scenario 5	0.7	0.5	50	50
Scenario 6	0.5	0.2	100	100

measure, and ranked third with a GD metric value of 0.0166. In contrast, MOEA, NSGA-2, and NSGA-3 demonstrate better values in Delta measure, with values 0.8698, 0.6869, and 0.6420, respectively. While in Generational distance the values of these methods 0.0471, 0.0019, and 0.0039, respectively.

MEPSOLA's superior performance in MOO scenario is a direct result of its innovative methodology that incorporates a multi-objective-aware criterion. This criterion is specifically designed to address the complexities and trade-offs between conflicting objectives effectively. MEPSOLA well employs a multi-exemplar selection process, which facilitates a concurrent balance between exploration and exploitation, thereby enhancing its convergence capabilities as well the diversity. Furthermore, the incorporation of Tabu search as a conditional local search mechanism within the algorithm for exemplar selection over a predetermined time frame significantly improves search efficiency and solution quality. By periodically updating the search strategy to circumvent previously visited or 'tabooed' solutions, the algorithm is able to escape local optima. This promotes exploration in new and potentially more optimal regions of the solution space. Such strategic application of Tabu search ensures dynamic adaptation to the evolving landscape of the search space, thereby improving the algorithm's capacity to identify high-quality, diverse solutions across multiple objectives. Moreover, the employment of a smart initialization strategy enhances the search effectiveness by ensuring that the initial population encompasses a broad and representative range of the solution space. This strategy prevents premature convergence on suboptimal regions, providing a solid foundation for the algorithm to efficiently explore diverse solutions.

FIGURE 9. **Parameters sensitivities for ZDT-1 function.**

Furthermore, MEPSOLA's superiority in SC, HV, and the NDS, along with sensible results in Delta and GD metrics. Further study for these measures should be made to understand the performance of the proposed method. The HV, quantifies the volume covered by the Pareto front, offering a holistic measure that captures both diversity and convergence [44]. SC complements HV by evaluating how well the Pareto front covers the objective space, ensuring the exploration of diverse solutions [48]. This dual focus on HV and SC provides a nuanced understanding of the algorithm's

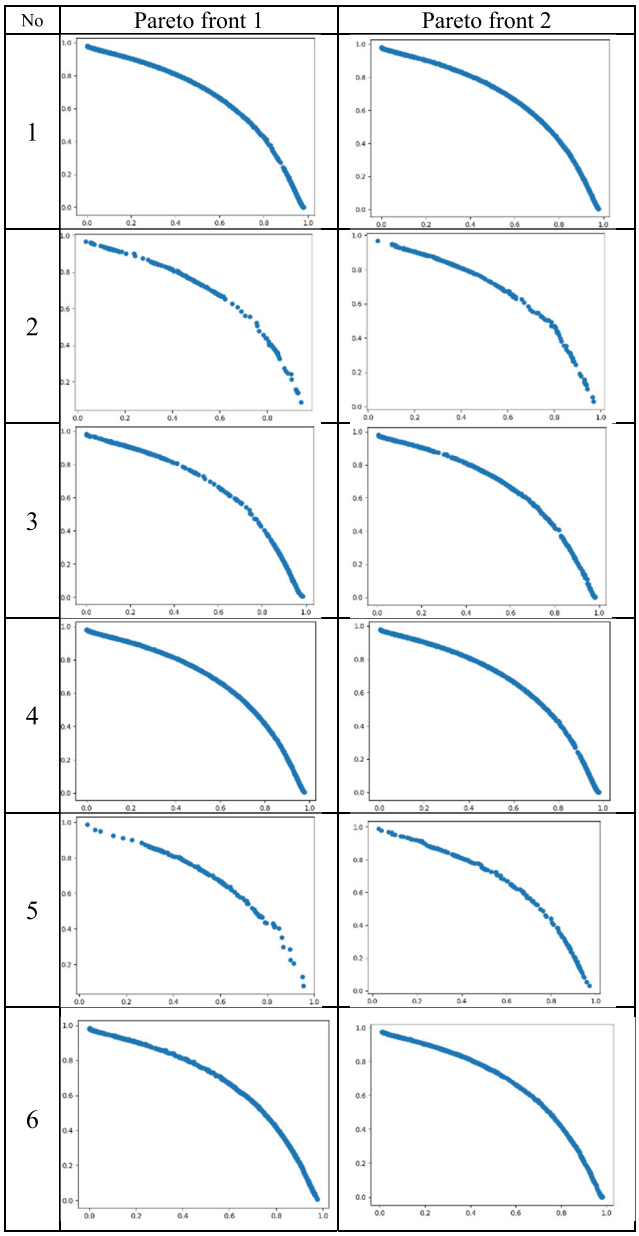

FIGURE 10. Parameters sensitivities for FON function.

performance in terms of both spread and representation. NDS, with its focus on diversity, avoidance of clustering, exploration of trade-offs, and robustness provision, emerges as a valuable metric in MOO. It ensures a thorough representation of the Pareto front, offering decision-making flexibility for users and facilitating benchmarking of MOO algorithms [49]. On the other hand, while Delta and GD are valuable measures, but they have limitations in providing a holistic view of the Pareto front. Delta focuses on specific aspects, such as spread relative to a reference set, while GD quantifies the average distance to the true Pareto front, potentially overlooking overall spread [50]. This highlights the need for a more comprehensive evaluation beyond these specific aspects. By considering NDS alongside HV and SC, a more well-rounded assessment of our algorithm performance is achieved, encompassing overall quality and diversity. This balanced approach enhances the understanding of how MOO algorithms navigate various objectives and trade-offs.

H. COMPLEXITY ANALYSIS

In this subsection, the computational complexity analysis of MEPSOLA is evaluated. This multi-objective optimization algorithm, based on a meta-heuristic approach, involves multiple computational steps, each contribute to the overall complexity.

The process of solutions initialization phase as illustrate in Table 7, involving equal range sampling, Cartesian product computation, and initialization of particle positions and velocities. The most computationally intensive part here is the Cartesian product due to an exponential complexity in the number of decision variables.

Next, the phase of exemplar selection for particles as demonstrated in Table 8, introducing a threshold-based repetition mechanism and a Tabu search for handling repeated selections, adding complexity based on the number of Tabu search iterations and neighborhood size.

Moreover, the conditional Tabu search further refines this process, iterating until a stopping criterion is met, as illustrated in Table 9.

Overall MEPSOLA algorithm complexity as shown in Table 10, including updates particle positions, velocities, costs, and exemplars, and continuously refining the repository of non-dominated particles. The overall complexity of the entire process combines these steps, with the main loop's complexity being the most significant due to its iterative nature and the number of operations per iteration. The final complexity reflects the combined effort of initializing the population, running the main optimization loop, and maintaining the repository of solutions. Based on the tables, the overall complexity of MEPSOLA is given as:

$$O\left(T \cdot \left(pop_{size} \cdot (n + C + M + E) + {pop_{size}}^2\right) + n + m^n \right.$$
$$\left. + pop\_size \cdot n + pop\_size \cdot C + {pop_{size}}^2 + pop\_size \cdot E\right)$$

I. PARAMETER SENSITIVITY

In parameters sensitivity we evaluated MEPSOLA algorithm and compared to the other benchmarks in different 8 scenarios based on ZDT-1 mathematical function and six different scenarios on FON mathematical function to evaluate the effect on Pareto front shape, these evaluations help in understanding how each parameter influences on the algorithm's performance, the most sensitive parameters were chosen that have direct impact on the performance of the proposed algorithm was the iteration of the MEPSOLA, population size PoP, inertia weight W, and the acceleration parameter c. The Table 11, illustrate the different scenarios parameters with the original setting before these scenarios.

From Fig. 9, it is observed that MEPSOLA performs robustly in many scenarios, particularly in dominance and Pareto front coverage. It suggests that initial parameter settings (as in the original scenario) might be near-optimal for MEPSOLA. The variation in performance across scenarios from 1 to 8 indicates sensitivity to certain parameter adjustments. Specifically, changes in the iteration count and population size appear to impact performance more significantly than changes in inertia weight W and the acceleration parameter c. NSGA-2 and NSGA-3 show varying degrees of competition with MEPSOLA, with NSGA-3 often providing competitive in some scenarios for set coverage or in Delta Measures, indicating that for some parameter settings, these algorithms might outperform MEPSOLA in uniform distribution and the quality of solutions.

Further parameter sensitivity analysis conducted on FON mathematical function on six scenarios as presented in the Table 12, to assess the effect of these parameters on the Pareto front shape in critical aspects like the diversity the convergence and the solutions distribution.

The results in Fig. 10, for FON function sensitivity to parameters reveals all scenarios were capable of producing a fair enough Pareto front with high similarity with the true Pareto. However, scenario 6 has produced the best solutions distributions Pareto with the least number of populations compare to 1and 4.

VI. CONCLUSION AND FUTURE WORKS

This article, introduces an innovative variant of MOPSO named Multi-Exemplar Particle Swarm Optimization with Local Awareness (MEPSOLA). MEPSOLA excels in solving complex multi-objective optimization scenarios through its innovative methodology, which incorporates a multi-objective-aware criterion designed to address the complexities and trade-offs between conflicting objectives effectively. MEPSOLA employs a multi-exemplar selection process, striking a balance between exploration and exploitation to enhance convergence capabilities and diversity. Additionally, the integration of Tabu search as a conditional local search mechanism significantly improves search efficiency and solution quality. By dynamically updating the search strategy, MEPSOLA can escape local optima and explore new, potentially more optimal regions of the solution space. it also addresses the issue of initialization sensitivity by incorporating equal sampling initialization enhancing enhance the algorithm's ability to effectively explore diverse regions of the solution space.

We compared MEPSOLA's performance against well-established benchmarks in the field, including MOEA, NSGA-2, and NSGA-3, across various mathematical functions. The results demonstrate MEPSOLA's superiority across majority of the evaluation measures. The superiority obvious in Set Coverage, showcasing dominance over all benchmarks with an average median percentage dominance across mathematical problems of 99.22%, 69%, and 93.58%, respectively. Moreover, MEPSOLA achieves higher hyper-volumes, with average percentage increases of 93.4%, 78.67%, and 93.66%, respectively.

MEPSOLA's innovative methodology, which integrates a multi-objective-aware criterion and a balanced exploration-exploitation approach, significantly enhances convergence capabilities and diversity. The incorporation of local search further boosts search efficiency, allowing for the escape from local optima and exploration of new regions. Comparative evaluations consistently highlight MEPSOLA's superiority over benchmarks across various functions, showcasing its adaptability and potential for integration with other fields.

Despite its promising outcomes, there is potential scope for future investigation to explore MEPSOLA across different domains and its adaptability to a wider range of real-world problems. Enhancing convergence consistency through hyperparameter refinement could further improve MEPSOLA's performance. Future work may involve exploring real-world applications such as feature selection and integrating MEPSOLA with other optimization techniques to provide deeper insights and advance the field.

ACKNOWLEDGMENT

In this article ChatGPT used for improve only the writing quality.

REFERENCES

[1] H. R. R. Zaman and F. S. Gharehchopogh, "An improved particle swarm optimization with backtracking search optimization algorithm for solving continuous optimization problems," *Eng. Comput.*, vol. 38, no. 4, pp. 2797–2831, Oct. 2022, doi: 10.1007/s00366-021-01431-6.

[2] M. S. Noori, R. K. Z. Sahbudin, A. Sali, and F. Hashim, "Feature drift aware for intrusion detection system using developed variable length particle swarm optimization in data stream," *IEEE Access*, vol. 11, pp. 128596–128617, 2023, doi: 10.1109/ACCESS.2023.3333000.

[3] M. H. Nadimi-Shahraki, S. Taghian, S. Mirjalili, L. Abualigah, M. A. Elaziz, and D. Oliva, "EWOA-OPF: Effective whale optimization algorithm to solve optimal power flow problem," *Electronics*, vol. 10, no. 23, p. 2975, Nov. 2021, doi: 10.3390/electronics10232975.

[4] G. H. M. Mendonça, F. G. D. C. Ferreira, R. T. N. Cardoso, and F. V. C. Martins, "Multi-attribute decision making applied to financial portfolio optimization problem," *Exp. Syst. Appl.*, vol. 158, Nov. 2020, Art. no. 113527, doi: 10.1016/j.eswa.2020.113527.

[5] S.-H. Huang, Y.-H. Huang, C. A. Blazquez, and C.-Y. Chen, "Solving the vehicle routing problem with drone for delivery services using an ant colony optimization algorithm," *Adv. Eng. Informat.*, vol. 51, Jan. 2022, Art. no. 101536, doi: 10.1016/j.aei.2022.101536.

[6] S. Arora and P. Anand, "Binary butterfly optimization approaches for feature selection," *Exp. Syst. Appl.*, vol. 116, pp. 147–160, Feb. 2019, doi: 10.1016/j.eswa.2018.08.051.

[7] M. Rostami, S. Forouzandeh, K. Berahmand, and M. Soltani, "Integration of multi-objective PSO based feature selection and node centrality for medical datasets," *Genomics*, vol. 112, no. 6, pp. 4370–4384, Nov. 2020, doi: 10.1016/j.ygeno.2020.07.027.

[8] A. Morales-Hernández, I. Van Nieuwenhuyse, and S. R. Gonzalez, "A survey on multi-objective hyperparameter optimization algorithms for machine learning," *Artif. Intell. Rev.*, vol. 56, no. 8, pp. 8043–8093, 2023, doi: 10.1007/s10462-022-10359-2.

[9] N. Saini and S. Saha, "Multi-objective optimization techniques: A survey of the state-of-the-art and applications: Multi-objective optimization techniques," *Eur. Phys. J. Special Topics*, vol. 230, no. 10, pp. 2319–2335, Sep. 2021, doi: 10.1140/epjs/s11734-021-00206-w.

[10] A. G. Gad, "Particle swarm optimization algorithm and its applications: A systematic review," *Arch. Comput. Methods Eng.*, vol. 29, no. 5, pp. 2531–2561, Aug. 2022, doi: 10.1007/s11831-021-09694-4.

[11] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. Int. Conf. Neural Netw.*, Nov. 1995, pp. 1942–1948, doi: 10.1109/icnn.1995.488968.

[12] J. Isacc Flores-Mendoza and E. Mezura-Montes, "Dynamic adaptation and multiobjective concepts in a particle swarm optimizer for constrained optimization," in *Proc. IEEE Congr. Evol. Comput.*, Jun. 2008, pp. 3427–3434, doi: 10.1109/CEC.2008.4631261.

[13] J. Yang, J. Zou, S. Yang, Y. Hu, J. Zheng, and Y. Liu, "A particle swarm algorithm based on the dual search strategy for dynamic multi-objective optimization," *Swarm Evol. Comput.*, vol. 83, Dec. 2023, Art. no. 101385, doi: 10.1016/j.swevo.2023.101385.

[14] K. Yasuda, A. Ide, and N. Iwasaki, "Adaptive particle swarm optimization," in *Proc. IEEE Int. Conf. Syst., Man Cybern. Conf.*, Oct. 2003, pp. 1554–1559.

[15] X. Liu, P. Zhang, H. Fang, and Y. Zhou, "Multi-objective reactive power optimization based on improved particle swarm optimization with ϵ-greedy strategy and Pareto archive algorithm," *IEEE Access*, vol. 9, pp. 65650–65659, 2021, doi: 10.1109/ACCESS.2021.3075777.

[16] B. Morales-Castañeda, D. Zaldívar, E. Cuevas, F. Fausto, and A. Rodríguez, "A better balance in metaheuristic algorithms: Does it exist?" *Swarm Evol. Comput.*, vol. 54, May 2020, Art. no. 100671, doi: 10.1016/j.swevo.2020.100671.

[17] G. Xu, Q. Cui, X. Shi, H. Ge, Z.-H. Zhan, H. P. Lee, Y. Liang, R. Tai, and C. Wu, "Particle swarm optimization based on dimensional learning strategy," *Swarm Evol. Comput.*, vol. 45, pp. 33–51, Mar. 2019, doi: 10.1016/j.swevo.2018.12.009.

[18] W. Huang and W. Zhang, "Adaptive multi-objective particle swarm optimization using three-stage strategy with decomposition," *Soft Comput.*, vol. 25, no. 23, pp. 14645–14672, Dec. 2021, doi: 10.1007/s00500-021-06262-7.

[19] W. Song and Z. Hua, "Multi-exemplar particle swarm optimization," *IEEE Access*, vol. 8, pp. 176363–176374, 2020, doi: 10.1109/ACCESS.2020.3026620.

[20] H. Dhahri, I. Rahmany, A. Mahmood, E. Al Maghayreh, and W. Elkilani, "Tabu search and machine-learning classification of benign and malignant proliferative breast lesions," *BioMed Res. Int.*, vol. 2020, pp. 1–10, Feb. 2020, doi: 10.1155/2020/4671349.

[21] A. K. Sangaiah and R. Khanduzi, "Tabu search with simulated annealing for solving a location-protection-disruption in hub network," *Appl. Soft Comput.*, vol. 114, Jan. 2022, Art. no. 108056, doi: 10.1016/j.asoc.2021.108056.

[22] *Particle Swarm Optimization—Goldsmiths Research Online*. Accessed: Sep. 25, 2023. [Online]. Available: https://research.gold.ac.uk/id/eprint/992/

[23] M. Clerc and J. Kennedy, "The particle swarm–explosion, stability, and convergence in a multidimensional complex space," *IEEE Trans. Evol. Comput.*, vol. 6, no. 1, pp. 58–73, Feb. 2002, doi: 10.1109/4235.985692.

[24] Y. Shi, "Particle swarm optimization: Developments, applications and resources," in *Proc. Congr. Evol. Comput.*, Seoul, (South) Korea, May 2001, pp. 81–86, doi: 10.1109/CEC.2001.934374.

[25] C. A. C. Coello, G. B. Lamont, and D. A. V. Veldhuizen, *Evolutionary Algorithms for Solving Multi-Objective Problems*, 2nd ed., 2007.

[26] K. Deb and J. Sundar, "Reference point based multi-objective optimization using evolutionary algorithms," in *Proc. 8th Annu. Conf. Genetic Evol. Comput.*, Jul. 2006, pp. 635–642, doi: 10.1145/1143997.1144112.

[27] B. Tran, B. Xue, and M. Zhang, "Variable-length particle swarm optimization for feature selection on high-dimensional classification," *IEEE Trans. Evol. Comput.*, vol. 23, no. 3, pp. 473–487, Jun. 2019, doi: 10.1109/TEVC.2018.2869405.

[28] B. Wang, Y. Sun, B. Xue, and M. Zhang, "Evolving deep convolutional neural networks by variable-length particle swarm optimization for image classification," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2018, pp. 1–8, doi: 10.1109/CEC.2018.8477735.

[29] K. Yin, B. Tang, M. Li, and H. Zhao, "A multi-objective optimization approach based on an enhanced particle swarm optimization algorithm with evolutionary game theory," *IEEE Access*, vol. 11, pp. 77566–77584, 2023, doi: 10.1109/ACCESS.2023.3298058.

[30] Y. Hu, Y. Zhang, and D. Gong, "Multiobjective particle swarm optimization for feature selection with fuzzy cost," *IEEE Trans. Cybern.*, vol. 51, no. 2, pp. 874–888, Feb. 2021, doi: 10.1109/TCYB.2020.3015756.

[31] V. Trivedi, P. Varshney, and M. Ramteke, "A simplified multi-objective particle swarm optimization algorithm," *Swarm Intell.*, vol. 14, no. 2, pp. 83–116, Jun. 2020, doi: 10.1007/s11721-019-00170-1.

[32] K. Alkebsi and W. Du, "A fast multi-objective particle swarm optimization algorithm based on a new archive updating mechanism," *IEEE Access*, vol. 8, pp. 124734–124754, 2020, doi: 10.1109/ACCESS.2020.3007846.

[33] H. Han, Y. Liu, Y. Hou, and J. Qiao, "Multi-modal multi-objective particle swarm optimization with self-adjusting strategy," *Inf. Sci.*, vol. 629, pp. 580–598, Jun. 2023, doi: 10.1016/j.ins.2023.02.019.

[34] D. C. Valencia-Rodríguez and C. A. Coello Coello, "Influence of the number of connections between particles in the performance of a multi-objective particle swarm optimizer," *Swarm Evol. Comput.*, vol. 77, Mar. 2023, Art. no. 101231, doi: 10.1016/j.swevo.2023.101231.

[35] H. Han, L. Zhang, A. Yinga, and J. Qiao, "Adaptive multiple selection strategy for multi-objective particle swarm optimization," *Inf. Sci.*, vol. 624, pp. 235–251, May 2023, doi: 10.1016/j.ins.2022.12.077.

[36] A. Aboud, N. Rokbani, R. Fdhila, A. M. Qahtani, O. Almutiry, H. Dhahri, A. Hussain, and A. M. Alimi, "DPb-MOPSO: A dynamic Pareto bi-level multi-objective particle swarm optimization algorithm," *Appl. Soft Comput.*, vol. 129, Nov. 2022, Art. no. 109622, doi: 10.1016/j.asoc.2022.109622.

[37] Y. Lu, B. Li, S. Liu, and A. Zhou, "A population cooperation based particle swarm optimization algorithm for large-scale multi-objective optimization," *Swarm Evol. Comput.*, vol. 83, Dec. 2023, Art. no. 101377, doi: 10.1016/j.swevo.2023.101377.

[38] Y. Cui, X. Meng, and J. Qiao, "A multi-objective particle swarm optimization algorithm based on two-archive mechanism," *Appl. Soft Comput.*, vol. 119, Apr. 2022, Art. no. 108532, doi: 10.1016/j.asoc.2022.108532.

[39] Y. Li, Y. Zhang, and W. Hu, "Adaptive multi-objective particle swarm optimization based on virtual Pareto front," *Inf. Sci.*, vol. 625, pp. 206–236, May 2023, doi: 10.1016/j.ins.2022.12.079.

[40] M.-F. Leung, C. A. C. Coello, C.-C. Cheung, S.-C. Ng, and A. K. Lui, "A hybrid leader selection strategy for many-objective particle swarm optimization," *IEEE Access*, vol. 8, pp. 189527–189545, 2020, doi: 10.1109/ACCESS.2020.3031002.

[41] A. Madani, A. Engelbrecht, and B. Ombuki-Berman, "Cooperative coevolutionary multi-guide particle swarm optimization algorithm for large-scale multi-objective optimization problems," *Swarm Evol. Comput.*, vol. 78, Apr. 2023, Art. no. 101262, doi: 10.1016/j.swevo.2023.101262.

[42] W. H. Bangyal, A. Hameed, W. Alosaimi, and H. Alyami, "A new initialization approach in particle swarm optimization for global optimization problems," *Comput. Intell. Neurosci.*, vol. 2021, pp. 1–17, May 2021, doi: 10.1155/2021/6628889.

[43] M. Zhang, H. Wang, Z. Cui, and J. Chen, "Hybrid multi-objective cuckoo search with dynamical local search," *Memetic Comput.*, vol. 10, no. 2, pp. 199–208, Jun. 2018, doi: 10.1007/s12293-017-0237-2.

[44] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002, doi: 10.1109/4235.996017.

[45] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, Part I: Solving problems with box constraints," *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 577–601, Aug. 2014, doi: 10.1109/TEVC.2013.2281535.

[46] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712–731, Dec. 2007, doi: 10.1109/TEVC.2007.892759.

[47] J. A. Nuh, T. W. Koh, S. Baharom, M. H. Osman, and S. N. Kew, "Performance evaluation metrics for multi-objective evolutionary algorithms in search-based software engineering: Systematic literature review," *Appl. Sci.*, vol. 11, no. 7, p. 3117, Mar. 2021, doi: 10.3390/app11073117.

[48] J. D. Knowles and D. W. Corne, "Approximating the nondominated front using the Pareto archived evolution strategy," *Evol. Comput.*, vol. 8, no. 2, pp. 149–172, Jun. 2000, doi: 10.1162/106365600568167.

[49] J. Bader and E. Zitzler, "HypE: An algorithm for fast hypervolume-based many-objective optimization," *Evol. Comput.*, vol. 19, no. 1, pp. 45–76, Mar. 2011, doi: 10.1162/EVCO_a_00009.

[50] E. Zitzler and L. Thiele, "Multiobjective optimization using evolutionary algorithms—A comparative case study," in *Parallel Problem Solving From Nature—PPSN V* (Lecture Notes in Computer Science), vol. 1498, A. E. Eiben, T. Bäck, M. Schoenauer, and H.-P. Schwefel, Eds., Berlin, Germany: Springer, 1998, pp. 292–301, doi: 10.1007/BFb0056872.

MUSTAFA SABAH NOORI received the B.Sc. degree in computer engineering from the University of Technology, Baghdad, Iraq, in 2006, and the M.Sc. degree in communication and computer engineering from Universiti Kebangsaan Malaysia (UKM), in 2016. He is currently pursuing the Ph.D. degree with the Department of Computer and Communication Systems Engineering, Faculty of Engineering, Universiti Putra Malaysia (UPM). His current research interests include the Internet of Things, intrusion detection systems, data mining, network communication, network security, machine learning, and cyber security.

RATNA K. Z. SAHBUDIN received the B.Sc. degree in electrical engineering from Fairleigh Dickinson University, Madison, NJ, USA, in 1988, the M.Sc. degree in RF and communication engineering from the University of Bradford, U.K., in 1992, and the Ph.D. degree in communication networks engineering from Universiti Putra Malaysia (UPM), Malaysia, in 2010. She is currently an Associate Professor with the Department of Computer and Communication Engineering, UPM. Her research interests include wireless communication, antenna, OCDMA, and optical communication.

ADUWATI SALI (Senior Member, IEEE) received the B.Eng. degree in electrical electronics engineering (communications) from The University of Edinburgh, U.K., in 1999, the M.Sc. degree in communications and network engineering from Universiti Putra Malaysia (UPM), Malaysia, in April 2002, and the Ph.D. degree in mobile and satellite communications from the University of Surrey, U.K., in July 2009. She was the Assistant Manager of Telekom Malaysia Bhd., from 1999 to 2000. She was the Deputy Director of the UPM Research Management Centre (RMC), responsible for research planning and knowledge management, from 2016 to 2019. She has been a Professor with the Department of Computer and Communication Systems, Faculty of Engineering, UPM, since February 2019. She is currently a Chartered Engineer (C.Eng.) registered under U.K. Engineering Council and a Professional Engineer (P.Eng.) under the Board of Engineers Malaysia (BEM). She is involved with IEEE as the Chair of ComSoc/VTS Malaysia, in 2017 and 2018, and the Young Professionals (YP), in 2015; and has been with Young Scientists Network-Academy of Sciences Malaysia (YSN-ASM), as an Honorary Member, since 2020, the Chair, in 2018, and the Co-Chair of Science Policy, in 2017. She was a recipient of the 2018 Top Research Scientists Malaysia (TRSM) Award from the Academy of Sciences Malaysia (ASM), the Finalist of Study UK Alumni Award 2020–2021 Professional Achievement Award, and the Honourable Mention and Regional Finalist for the 7th Annual Underwriters Laboratories-ASEAN-U.S. Science Prize for Women 2021.

FAZIRULHISYAM HASHIM (Member, IEEE) received the B.Eng. degree from Universiti Putra Malaysia, in 2002, the M.Sc. degree from Universiti Sains Malaysia, in 2006, and the Ph.D. degree from The University of Sydney, Australia, in 2010. He is currently an Associate Professor with the Department of Computer and Communication Systems Engineering, Universiti Putra Malaysia. He also leads the Communication Network Laboratory and is a member with the Wireless and Photonics Networks (WiPNET) Research Group. He has published more than 150 papers in reputable journals and conferences. His research interests include wireless communication networks, including mobile networks, network and computer security, wireless sensor networks, software-defined networking, network function virtualisation, blockchain, and vehicular communication.