IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 20 June 2023, accepted 5 July 2023, date of publication 10 July 2023, date of current version 16 October 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3294095

SURVEY

Many-Objective Real-World Engineering Problems: A Comparative Study of State-of-the-Art Algorithms

VIKAS PALAKONDA AND JAE-MO KANG, (Member, IEEE)
Department of Artificial Intelligence, Kyungpook National University, Daegu 41566, South Korea

Corresponding author: Jae-Mo Kang (jmkang@knu.ac.kr)

This work was supported in part by the National Research Foundation of Korea (NRF) Grant funded by the Korea Government through the Ministry of Science and ICT (MSIT) under Grant 2022R1A4A1033830; in part by MSIT, South Korea, through the Information Technology Research Center (ITRC) Support Program supervised by the Institute of Information and Communications Technology Planning and Evaluation (IITP) under Grant IITP-2023-2020-0-01808; and in part by the National Research Foundation (NRF), South Korea, under Project BK21 FOUR.

ABSTRACT Many-objective optimization has recently gained popularity as it poses significant challenges for the existing algorithms. Therefore, numerous optimization algorithms have been developed to handle many-objective optimization in the literature. In addition, several studies have conducted experimental comparisons to assess the performance of optimization algorithms. Nevertheless, existing empirical studies have analyzed the performance of optimization algorithms on well-defined test problems, but it remains unclear whether the results translate to real-world scenarios. Furthermore, empirical studies on validating the performance of algorithms on real-world many-objective problems are intriguing but not yet fully explored. Therefore, in this article, we present a comprehensive comparative study evaluating the performance of 15 state-of-the-art algorithms on ten real-world many-objective applications with four to ten objectives from various domains. Further, these ten applications exhibit various mathematically challenging properties, including stochastic objectives, complex Pareto frontiers, and strong nonlinearity. In addition, four performance metrics are employed to visualize the performance of MOEAs in experimental settings. Based on comparative results, the performance of state-of-the-art algorithms with respect to different problems is evaluated herein.

INDEX TERMS Convergence, diversity, many-objective optimization, multi-objective optimization, real-world application.

I. INTRODUCTION

Multi-objective optimization problems (MOPs) are ubiquitous in real-world applications and typically involve multiple conflicting objectives that require simultaneous optimization [1]. MOPs are fundamental in various domains and applications, including scheduling [2], deep learning [3], machine learning [4], [5], and other industrial applications [6]. MOPs are typically expressed as follows:

$$\begin{aligned}&Minimize\ \boldsymbol{F}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_M(\boldsymbol{x}))^T\\&subject\ to\ \boldsymbol{x} \in \boldsymbol{\Omega},\end{aligned} \tag{1}$$

where M is the number of objectives considered, and x represents an n-dimensional decision vector. Moreover, MOPs comprise conflicting objectives; hence improving one objective without deteriorating at least one or other objectives is impossible [6]. Thus, instead of a single optimal solution, a set of nondominated solutions known as Pareto-optimal solutions must be developed for solving MOPs. A collection of Pareto-optimal solutions is called a Pareto-optimal set (PS) in decision space, and its projection on an objective space is referred to as a Pareto front (PF). Many-objective optimization problems (MaOPs) are a particular category of MOPs having more than three objectives [7]. Similar to MOPs, many practical problems, such as time series

The associate editor coordinating the review of this manuscript and approving it for publication was Bijoy Chand Chatterjee.

learning [8], engineering design [9], and ensemble learning [10], [11] involve MaOPs.

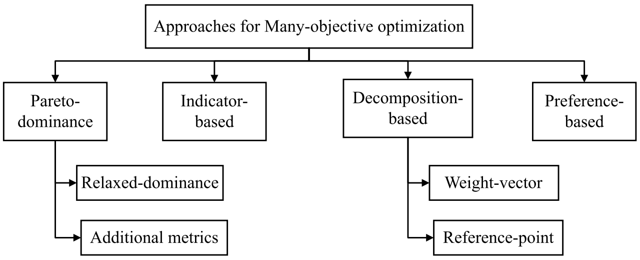

FIGURE 1. Approaches for many-objective optimization.

Multi-objective evolutionary algorithms (MOEAs) using a population-based iterative search engine effectively solve MOPs and MaOPs [12]. MOEAs enable the approximation of PF by constructing well-converged and well-distributed nondominated solutions. [13]. In other words, MOEAs aim to achieve two conflicting but critical goals, namely, convergence and diversity. Given the robust search process, MOEAs achieved better performance for MOPs than the MaOPs. In addition to MOEAs, other optimization algorithms such as particle swarm optimization (PSO) [14], [15], firefly algorithm (FA) [16], act colony optimization (ACO) [17], etc. have been developed to address MaOPs. However, when dealing with MaOPs, algorithms encounter three main challenges [18]. First, as the number of objectives increases, the nondominated solutions increase, resulting in the weakened selection pressure toward PF. Second, achieving convergence and diversity in MaOPs is extremely challenging owing to their high dimensionality. Third, the effect of evolutionary operators in generating promising offspring solutions is considerably reduced due to the high dimensionality in MaOPs [19]. Consequently, considerable research efforts have been expended to develop efficient algorithms and technologies to address the scalability issues of optimization algorithms.

Based on the selection mechanism, the optimization algorithms can be roughly classified into a) Pareto-dominance approaches [20], [21], b) indicator-based approaches [22], [23], [24], [25], c) decomposition-based approaches [26], and d) preference-based approaches [27], [28]. Fig. 1 illustrates different categories of optimization algorithms proposed for solving MaOPs. However, each class of optimization criteria has its disadvantages in solving MaOPs. In Pareto-dominance-based approaches [20], [21], the nondominated sorting procedure is adopted to differentiate candidate solutions according to their dominance relation. The major issue with Pareto-dominance-based approaches is relatively low selection pressure toward the true PF because many solutions are nondominated in a high-dimensional objective space. To overcome this issue, two modifications to Pareto-dominance-based approaches were proposed as depicted in Fig. 1, a) Pareto-dominance-based approaches with additional convergence metrics [13], [29] and b) relaxed dominance-based approaches [30], [31]. However, additional convergence metrics are associated with computational complexity and relaxed dominance often gets trapped in local optima. The indicator-based approaches transform the MOPs/MaOPs into a problem of optimizing an indicator using a performance metric, such as the epsilon indicator [22], hypervolume indicator [23], etc. The critical shortcoming of indicator-based approaches is the high computational complexity of some performance indicators, such as hypervolume (HV). The decomposition-based approaches [26], [32], [33] divide the MOPs/MaOPs into many subproblems and collaboratively optimize them. Decomposition-based approaches are highly sensitive to the shape of PF, and the need to specify a set of well-distributed reference points/weight vectors is another disadvantage of decomposition-based approaches. Preference-based approaches provide decision-makers with a small set of representative solutions in the region of interest according to their preferences. However, in preference-based approaches, articulating correct preferences is extremely difficult [6].

Most optimization algorithms developed for MaOPs have been tested on well-defined test problems to assess their efficiency and effectiveness. These test problems comprise a benchmark suite designed to reflect challenging problem characteristics, including but not limited to multimodality, discreteness, nonconvexity, deception, nonuniformity, isolated optima, non-separability, and scalability of the number of objectives as well as decision variables [34]. Although these benchmarks capture various challenging characteristics, whether they can capture the complexity observed in real-world applications remains to be seen [35]. These numerically-defined test problems cannot adequately capture the intricate details of complex real-world problems. This is an important and significant concern. Moreover, testing solely on numerically-defined test problems could result in a myopic understanding of the translational value of MOEAs in the context of real-world applications [36].

Many comparative studies in the past analyzed the performance of optimization algorithms [37]. However, these were often restricted to artificial test problems, and only a few studies focused on using MOEAs for real-world applications [34], [35], [36]. Unfortunately, the existing studies on real-world applications are confined to just a few approaches to solving two or three real-world problems. Furthermore, no existing study provides a comparative analysis of the performance of MOEAs on real-world multi-objective problems. Therefore, in this article, we present a thorough comparative study on the performance of MOEAs in solving real-world problems from different domains herein.

The main contributions of this article are summarized as follows:

- Herein, we compare the performance of 15 state-of-the-art algorithms in solving ten real-world many-objective problems from different domains to demonstrate the advantages of optimization algorithms over other algorithms and encourage domain experts to share their challenges with researchers.

- The performance of MOEAs is analyzed based on four performance indicators, including hypervolume (HV), inverted generational distance (IGD), generational distance (GD), and pure diversity (PD).

The rest of this paper is organized as follows. Section II presents the research methodology, basic definitions related to the evolutionary multi-objective community, description of MOEAs, and real-world problems considered in this study. Section III presents the experimental results and discussions, and Section IV concludes the paper.

II. RESEARCH METHODOLOGY

A. EXPERIMENTAL FRAMEWORK

In this comparative study, we compare the performance of optimization algorithms from different selection categories in solving ten real-world many-objective problems. The experimental methodology is described as follows:

1) First, each algorithm is simulated on ten real-world many-objective problems for 30 individual runs, and the resultant approximated solution sets are recorded. Then, the number of generations is set as 250 for all real-world many-objective problems.
2) Next, because the true PF is not known for real-world problems, we create reference sets by combining well-known approximations of PF across all runs of all algorithms on a given problem.
3) Herein, we use some proper performance metrics to measure convergence and diversity, such as HV and IGD, respectively. In addition, GD and PD are employed to measure convergence and diversity, respectively.
4) Finally, the performance of algorithms in solving each real-world problem is analyzed based on the aforementioned performance metrics.

B. BASIC DEFINITION AND CONCEPTS

Definition 1: For any two solutions, $x_1 \in \Omega$ and $x_2 \in \Omega$, where Ω is the decision space, and solution x_1 dominates x_2 only if, $f_m(x_1) \leq f_m(x_2)$, $\forall m = 1, 2, \ldots, M$, and there exists $i = \{1, 2, .., M\}$ and $f_i(x_1) < f_i(x_2)$.

Definition 2: A decision vector, $x_1 \in \Omega$, is considered Pareto-optimal when no other decision vector, $x_1^* \in \Omega$ that dominates x_1, i.e., $x_1 \leq x_1^*$ exists.

Definition 3: The set comprising Pareto-optimal solutions is referred to as the Pareto-optimal set (PS), and the projection of the PS onto the objective vector space is termed PF.

C. ALGORITHM DESCRIPTION

In this comparative study, we have evaluated the efficacy of 15 state-of-the-art algorithms on ten real-world engineering problems. The selected algorithms belong to diverse categories of methodologies that have been suggested for addressing multi/many-objective optimization problems. The general working principles of the 15 state-of-the-art algorithms are outlined below.

1) *An MOEA with angle-based selection and shift-based density estimation* (AnD)[1] [38]: In the AnD algorithm, the angle-based selection strategy is employed along with shift density estimation to solve MaOPs. First, angle-based selection identifies a pair of individuals with the minor vector angle, which indicates that their search directions are similar. Next, the shift-based estimation strategy is used to determine inferior individuals based on convergence and diversity. Hence, the AnD algorithm uses a simple framework to remove inefficient individuals one by one.
2) *Ensemble fitness ranking with ranking restriction* (EFR-RR)[2]: EFR-RR is an improved version of ensemble fitness ranking (EFR) [39] with a ranking restriction scheme. EFR is an extension of maximum rank and average ranking [40] with highly generalized fitness aggregation) functions. EFR adopts the nondominated sorting genetic algorithms II (NSGAII) framework with considerable differences in environmental selection. These considerable differences are that instead of objective functions, EFR adopts fitness functions such as L_p-norm, Tchebycheff function, and Penalty-Boundary Intersection (PBI), which help EFR to achieve a balance between convergence and diversity in solving MaOPs.
3) *MOEA based on an ISDE+ indicator* (ISDE+)[3] [24]: ISDE+ algorithm proposes an indicator that effectively combines the advantages of the sum of objectives and shift-based density estimation (SDE). The sum of objectives increases the selection pressure towards the PF, and SDE preserves diversity among population members. Moreover, the ISDE+ indicator considers diversity estimation only for individuals with high converging abilities.
4) *An Indicator-based MOEA with boundary protection* (MaOEA-IBP)[4] [1]: The MaOEA-IBP algorithm proposes the worst-elimination strategy assisted by a boundary protection mechanism to improve the convergence, diversity, and coverage of a population. First, a pair of candidate solutions from the population with the smallest I_ϵ^+ values are identified and compared based on their indicator value. Subsequently, the solution with a higher I_ϵ^+ value is eliminated. If both have the same I_ϵ^+ value, then one solution is eliminated based on the boundary protection strategy.

[1] The source code for AnD algorithm is downloaded from https://intleo.csu.edu.cn/publication.html

[2] The source codes for the EFR-RR, MOEAD-URAW, NSGAIII, NSGAII-SDR, PiCEA-g, RPD-NSGAII, RVEA, SPEAR, t-DEA, NMPSO algorithms are downloaded from https://github.com/BIMK/PlatEMO

[3] The source code for ISDE+ algorithm is downloaded from https://github.com/P-N-Suganthan/CODES

[4] The source code for MaOEA-IBP algorithm is downloaded from https://github.com/CIA-SZU

5) *MOEA/D with uniformly randomly adaptive weights* (MOEAD-URAW)[2] [33]: The MOEAD-URAW algorithm generates subproblems by implementing uniformly random initialization and ensuring that the population size is flexible even with MaOPs. In addition, during the evolutionary process, MOEAD-URAW includes and removes subproblems using a sparsity-level function of the population. Further, weights are adapted in the MOEAD-URAW algorithms to deal with PFs with different shapes.
6) *Nondominated sorting genetic algorithm III* (NSGAIII)[2] [29]: NSGAIII follows the NSGA-II [20] framework with a significant modification in environmental selection. NSGA-II adopts Pareto-dominance and crowding distance to select the fittest candidate solutions. In NSGAIII, crowding distance is replaced with a set of well-spread reference points to maintain individual diversity. First, each individual candidate in the population is associated with a reference point based on its perpendicular distance to a reference line. Next, nondominated individual candidates close to the reference points are given more preference as compared to those farther away from the reference line.
7) *A strengthened dominance relation based many-objective evolutionary algorithm* (NSGAII-SDR)[2] [30]: The NSGAII-SDR algorithm proposes a strengthened dominance relation (SDR) to attain a trade-off between convergence and diversity among the population members. SDR adopts a niche technique developed based on the angles between candidate solutions. Each niche technique maintains the best-converged candidate solution within the niche technique.
8) *MOEA based on a pivot solution based selection* (Pi-MOEA)[5] [41]: The Pi-MOEA algorithm proposes a pivot solution based selection mechanism assisted by an adaptive neighborhood technique to solve MOPs and MaOPs. Pi-MOEA algorithm designs an adaptive neighborhood based on average rank to identify pivot solutions in each nondominated front. The pivot solutions drive the selection process toward convergence. In addition, Euclidean distance-driven density estimation is adopted in Pi-MOEA to enforce diversity.
9) *Preference-inspired coevolutionary algorithm with goals* (PICEA-g)[2] [27]: PICEA-g uses the preference-inspired coevolutionary algorithm (PICEA) to coevolve candidate solutions and decision-maker preferences. Adaptation through coevolution is noteworthy because harnessing its power for optimization is challenging (as opposed to exploring coevolutionary dynamics). Moreover, PiCEA-g aims to help decision-makers approximate the entire PF through posterior optimization.
10) *Decomposition-Based NSGA-II* (RPD-NSGAII)[2] [12]: In the RPD-NSGA-II approach, reference point-based dominance is proposed (RP-dominance), which compares individual candidates associated with different reference points. The RP-dominance framework combines the features of Pareto-dominance and decomposition-based approaches. Furthermore, two penalty-based boundary intersection (PBI) distances are adopted to achieve a trade-off between convergence and diversity.
11) *A Reference vector guided evolutionary algorithm* (RVEA)[2] [42]: The RVEA algorithm proposes angle-penalized distance (APD) to balance convergence and diversity. APD measures the convergence criterion using the distance between individuals and the ideal points, as well as the diversity criterion using the angle between population members and reference points. In addition, an adaptive strategy is used to adjust the distribution of reference points based on the range of objective functions.
12) *Strength Pareto evolutionary algorithm based on reference direction* (SPEA-R)[2] [32]: The SPEA-R algorithm partitions objective space into different subregions using a predefined set of reference directions. Further, candidate solutions in subregions are guided toward the predefined search directions. In addition, a diversity-first-convergence-second selection strategy increases the selection pressure adopted in SPEA-R. Finally, a restricted mating selection scheme is adopted to generate excellent offspring solutions.
13) *t-dominance based evolutionary algorithm* (t-DEA)[2] [31]: In t-DEA, a new dominance relation termed θ-dominance allocates the candidate solutions to different clusters represented by well-distributed reference points. Candidate solutions within the same cluster possess a competitive relationship based on a fitness function similar to penalty-based boundary intersection (PBI). The environmental selection based on θ-dominance selects candidate solutions with high fitness values in each cluster that ensure convergence and diversity.
14) *Novel Fitness mechanism based particle swarm optimization* (NMPSO)[2] [14]: In the NMPSO algorithm, a balanced fitness estimation (BFE) technique was proposed to address MaOPs. This BFE method incorporates convergence and diversity distances to alleviate the curse of dimensionality in MaOPs and direct particles to their real PFs. NMPSO also utilized two additional operators: the evolutionary search on the external archive and the novel velocity update equation. The evolutionary search is intended to overcome the ineffectiveness of PSO-based search on certain categories of MaOPs by providing an alternative search pattern. The novel velocity update equation, on the other hand, provides another PSO-based search direction and increases diversity.

[5] The source code for Pi-MOEA algorithm is downloaded from https://github.com/Vikas11475/Pi-MOEA

15) *Multi-objective Firefly algorithm (MOFA)*[6] [16]: In the MOFA approach, a learning-based search mechanism is adopted for firefly algorithm to tackle the continuous optimization problems.MOFA initializes a population of n fireflies so that they are distributed as uniformly as feasible throughout the search space using sampling techniques based on uniform distributions. After defining the tolerance or number of iterations, the iterations begin by assessing the brightness or objective values of all fireflies and comparing each pair. Then, a random weight vector is constructed with a sum of 1 to find the optimum solution.

Among the fifteen state-of-the-art algorithms considered in this study, NSGAII-SDR [30] and t-DEA [31] algorithms are relaxed dominance-based methods. The AnD [38], Pi-MOEA [41], and NSGAIII [29] belong to the category of algorithms that incorporate additional selection metrics. ISDE+ [24] and MaOEA-IBP [1] are approaches based on indicators. EFR-RR [39] and MOEAD-URAW [33] are decomposition-based methods that adapt weight vectors. RPD-NSGAII [12], RVEA [42], and SPEA-R [32] are algorithms that belong to the decomposition methods with reference-point adaptations. PiCEA-g [27] algorithm belongs to the category of preference-based approaches. NMPSO [14] and MOFA [16] are examples of swarm intelligence algorithms.

D. PROBLEM DESCRIPTION

In this comparative study, we consider ten famous real-world engineering problems with 4 to 10 objectives.

1) *Car side impact design problem* **(CSID)**[7]: The CSID [43] is a constrained three-objective problem that aims to optimize the vehicle side impact crashworthiness. CSID problem involves seven decision variables describing the thickness of the B-pillars, roof rail, door beam, cross-members, floor, etc. As suggested in [44], we take into account the unconstrained four objective CSID problem that aims to optimize car weight, the public force experienced by passengers, B-pillar responsible average velocity, and summation of the constraints.
2) *Conceptual marine design* **(CMD)**[7]: The CMD [45] problem involves six decision variables, three objectives, and nine constraints. The decision variables represent block coefficients, speed, draft, depth, beam, and length. As suggested in [44], we consider four objectives aimed at optimizing transportation cost, light-ship weight, the annual cargo transport capacity, and summation of the constraints to solve the CMD problem.
3) *Vehicle vibration model problem* **(VVM)**[8]: The VVM [46] problem consists of seven decision variables and five objectives. The decision variables are related to the seat stiffness coefficient, vehicle suspension stiffness coefficient, seat damping coefficient, vehicle suspension damping coefficient, and seat position in relation to the center of mass, respectively [46]. The five-objective VVM problem [47] considered herein aims at optimizing vertical seat acceleration, forward tire vertical velocity, rear tire vertical velocity, the relative displacement between forward tires and sprung mass, as well as, the relative displacement between rear tires and sprung mass.
4) *Location of a pollution monitoring system* **(LPMS)**[9] [48]: This problem aims to determine the location of a pollution monitoring system (LPMS) in two-dimensional decision space. This problem consists of five objectives that correspond to the expected information loss based on the estimates of five experts. As a result, finding a location that perfectly balances the five possible losses is needed. The problem formulation for LPMS is adopted from a previous study [49].
5) *Machining problem* **(MP)**[10] [50]: This problem considers the optimization of multiple criteria for machining operations on B390 die-cast aluminum alloy using VC-3 carbide tools. The MP consists of three decision variables, four objectives, and three constraints. The decision variables are related to the depth of cut, feed rate, and cutting speed. In this paper, we consider five-objective machining problems that aim at optimizing surface roughness, surface integrity, tool life, metal removal rate, and summation of constraints.
6) *Water resource planning* **(WRP)**[7] [51]: This problem entails the process of planning, developing, managing, and allocating water resources to maximize their use. The WRP problem consists of three decision variables related to maximum overflow rate, maximum treatment rate, and local detention storage capacity. In addition, the WRP problem consists of five objectives and seven constraints. As suggested in a previous study [44], we consider a six-objective WRP problem that optimizes drainage network cost, storage facility cost, treatment facility cost, expected flood damage cost, expected economic loss due to floods, and summation of constraints.
7) *Work roll cooling design problem* **(WRCD)**[10]: The WRCD problem [52] mainly focuses on shaping a metal surface by decreasing its thickness and creating a uniform surface. The WRCD problem consists of seven decision variables related to delay time, roll temperature, roll speed, cooling HTC, roll/stock contact length, stock temperature, and roll/stock contact HTC. In addition, WRCD consists of six objectives that aim

[6] The source code for MOFA algorithm is downloaded from https://github.com/zaleman/MOFA

[7] The source codes for the CSID, CMD, RWP, CCD problems are downloaded from https://ryojitanabe.github.io/reproblems/

[8] The source code for the VVM problem is provided by the authors.

[9] The source code for the LPMS problem is downloaded from https://www.mathworks.com/matlabcentral/fileexchange/87262-preference-based-indicator-mode?s_tid=prof_contriblnk

[10] The source codes for the MP and WRCD problems are formulated from https://link.springer.com/chapter/10.1007/978-3-319-54157-0_14

at optimizing change in temperature at roll surface, radial stress at the roll surface, change in temperature at 9mm depth, radial stress at 9mm depth, changes in temperature at 15mm depth, and radial stress at 15mm depth.

8) *Car cab design* **(CCD)**[7] *problem*: The CCD [44] problem considered herein consists of eleven decision variables. The decision variables are divided into two categories, where the first seven variables indicate the thickness of the B-pillar inner, B-pillar reinforcement, floor side inner, cross members, door beam, door beltline reinforcement, and rail roof, respectively. The remaining variables are stochastic in nature and represent the material of the B-pillar inner, floor side inner, barrier height, and barrier hitting position. As suggested in the previous work [44], we consider a nine-objective CCD problem with objectives as $f_1(x)$ (weight of the car), and the remaining eight objectives $(f_2(x), \ldots, f_9(x))$ correspond to each of the eight constraints.

9) *Radar waveform design* **(RWD)**[11] *problem*: The RWD [53] problem comprises 4-12 decision variables and nine objectives. Herein, we considered an unconstrained nine-objective RWD problem. Among the nine objectives, f_1 to f_8 objectives need to be maximized, and hence they are multiplied by -1 in order to convert them for minimization.

10) *General aviation aircraft* **(GAA)**[12] *design problem* [54]: The GAA design problem consists of nine decision variables and ten objectives. The design variables are related to tapering ratio, tail length ratio, seat width, engine activity factor, wing loading, propeller diameter, sweep angle, aspect ratio, and cruise speed. The GAA problem with ten objectives aims to optimize takeoff noise, empty weight, cruise speed, lift ratio, flight range, product family dissimilarity, purchase price, fuel weight, ride roughness, and direct operating cost.

The real-world problem considered in this paper consists of continuous variables, and the Pareto front is unknown to them. In addition, these ten applications exhibit a variety of mathematically difficult characteristics, such as stochastic objectives, complex Pareto frontiers, and robust nonlinearity. Among the ten problems, car side impact design problem, vehicle vibration model problem, machining problem, and car cab design problem fall into the category of mechanical design problems. The problems of conceptual marine design, water resource planning, and work roll cooling design fall under the category of process, design, and synthesis problems. The location of a pollution monitoring system is a chemical engineering problem, whereas the design of radar waveform and general aviation aircraft (GAA) are electronic design problems.

E. PERFORMANCE EVALUATION METRICS

We employ four performance metrics, namely, HV [55], IGD [56], GD [57], and PD [58] to evaluate and compare the quality of nondominated solution sets obtained from each MOEA. The HV and IGD metrics related to the convergence and diversity, respectively, of the final solution set obtained from each MOEA. The GD and PD metrics reflect convergence and diversity separately. These performance metrics are described as follows:

1) *Hypervolume* **(HV)**: The HV metric [55] estimates the volume of objective space occupied by the solution set obtained by each algorithm. The hypervolume metric is evaluated as

$$HV(S) = vol\left(\bigcup_{x\in S}[f_1(x), z_1] \times \ldots \times [f_M(x), z_M]\right), \tag{2}$$

where $vol(\cdot)$ indicates the Lebesgue measure, S is the solution set obtained by each MOEA, $z^r = (z_1, \ldots, z_M)^T$ specifies a reference point. As shown in Equation (2), a reference point is required to compute the HV metric. To this end, first, the objectives are normalized within the range of [0,1], and $(1, 1, \ldots 1)^M$ is set as a reference point. In addition, we have employed the Monte Carlo sampling method with 1,000,000 points to estimate the HV metric.

2) *Inverted generational distance* **(IGD)**: The IGD metric [56] measures the average distance between the uniformly sampled reference set and PF approximated by the final solution set obtained from each MOEA. The IGD metric can be evaluated as follows:

$$IGD(S, R) = \frac{\sum_{r\in R} d(r, S)}{|R|} \tag{3}$$

where S is the solution set achieved from each MOEA, R is the reference set, and $d(r, S)$ is the minimum Euclidean distance between each point in the reference set, r and the corresponding points in S. Herein, we consider 8000 uniformly sampled points as the reference set.

3) *Generational distance* **(GD)**: The GD metric measures the distance of the PF approximated using the obtained solution set with respect to the true PF. The GD metric is evaluated as follows:

$$GD(S, R) = \frac{\sqrt{\sum_{i=1}^{|S|} d_i^2}}{|S|} \tag{4}$$

where S is the solution set achieved from each MOEA, and d_i^2 is the Euclidean distance between each solution in the obtained solution set and its nearest point in the reference set.

4) *Pure diversity* **(PD)**: The PD metric measures the diversity of the solution set and is evaluated as follows:

$$PD(S) = \max_{x\in S}(PD(S \setminus \{p\}) + \min_{y\in S\setminus\{p\}} \|f(p) - f(q)\|_p) \tag{5}$$

[11] The source code for the RWD problem is downloaded from http://code.evanhughes.org/

[12] The source code for the GAA problem is downloaded from https://github.com/mariocastrogama/GAA-problem-MATLAB

$\|f(p) - f(q)\|_p$ denotes the L_p norm distance between individuals p and q in the objective space.

Note that the higher values of HV and PD metrics indicate better performance of MOEAs; in contrast, lower values of IGD and GD result in high performance of MOEAs.

III. EMPIRICAL RESULTS

We analyzed the performance of MOEAs based on their simulation runs for solving ten real-world many-objective problems based on the experimental settings employed herein.

A. EXPERIMENTAL SETTINGS

1) **Population size**: The ten real-world problems considered in this paper comprise 4-10 objectives. Hence, the population size is set as 120, 126, 132, 112, 156, 210, and 275 for 4-, 5-, 6-, 7-, 8-, 9-, and 10-objective problems, respectively.
2) **Number of function evaluations**: The number of function evaluations is set to 30,000, 31,500, 33000, 28000, 39000, 52500, and 68,750 for 4-, 5-, 6-, 7-, 8-, 9-, and 10-objective problems, respectively.
3) **Algorithm parameter settings**: The parameter settings for the MOEAs selected for comparison are adapted from their original publications. In the EFR-RR algorithm, the number of closest weight vectors is set to 2. In MOEAD-URAW, the number of candidate solutions replaced by every offspring solution is set to 2, and the probability of selecting parents is set to 0.9. In PICEA-g, the number of goals is set to $100 \times M$, where M is the number of objectives. In RVEA, the parameter controlling the rate of change of penalty is set to 2, and the frequency of employing reference vector adaptation is set to 0.1. In MOFA, the initial randomness factor is set to 0.2. The remaining algorithms are parameter-free.
4) **Genetic operators**: The Genetic operators, such as simulated binary crossover (SBX) [59] and polynomial mutation [60] used in MOEAs considered in this comparative study. The distribution index and probability for SBX are set as $n_c = 20$ and $p_c = 1.0$. The distribution index and probability for polynomial mutation are set as $n_m = 20$ and $p_m = 1/n$, respectively, where n is the number of decision variables.

B. EXPERIMENTAL RESULTS

The mean and standard deviation values of HV, IGD, GD, and PD metrics obtained for the MOEAs simulation runs corresponding to the ten real-world problems are listed in Tables 1-10. Furthermore, we displayed the ranking of each MOEA based on each performance metric in these tables. We highlight the best-performing MOEA for each real-world problem in terms of every performance metric. Further, to analyze the performance of MOEAs, we perform the Friedman test to rank the algorithms on each real-world problem and achieve an average rank.

- ***CSID problem***: The performance metric results for 15 algorithms on the CSID problem are listed in Table 1. The results presented in Table 1, demonstrate that in terms of HV and IGD metrics MaOEA-IBP algorithm performs better than other MOEAs followed by the NMPSO and ISDE+ algorithms. In terms of HV metric, NSGAII-SDR, MOFA, RVEA, and SPEA-R algorithms perform poorly on the CSID problem. Regarding the IGD metric, PiCEA-g, NSGAII-SDR, SPEA-R, and RVEA algorithms exhibit poor performance on the CSID problem. The NMPSO algorithm performs better in terms of GD metric, followed by the Picea-g, ISDE+, and NSGAII-SDR algorithms. By contrast, the Pi-MOEA, RVEA, EFR-RR, and RPD-NSGAII algorithms perform poorly on the CSID problem in terms of GD metric. In terms of the PD metric, the AnD algorithm performs better than other MOEAs followed by MOEAD-URAW, MOFA, and Pi-MOEA algorithms. The PiCEA-g, t-DEA, RVEA, and EFR-RR algorithms perform badly on the PD metric. Therefore, the MOEA-IBP, ISDE+, NMPSO, MOEAD-URAW, and AnD algorithms perform consistently better on all the performance metrics compared to RVEA, SPEA-R, and t-DEA algorithms.
- ***CMD problem***: The results of HV, IGD, GD, and PD metrics obtained for 15 algorithms are listed in Table 2. From the results presented in Table 2, the ISDE+ algorithm achieves superior performance than other MOEAs, followed by MOEAD-URAW, NMPSO, and MaOEA-IBP algorithms according to the HV metric. However, PiCEA-g, RVEA, NSGAIII, and NSGAII-SDR algorithms perform poorly on the CMD problem with respect to the HV metric. Regarding the IGD metric, MOEAD-URAW, ISDE+, Pi-MOEA, and MaOEA-IBP algorithms provide better performance than PiCEA-g, RVEA, NSGAII-SDR, and NSGAIII algorithms. In terms of the GD metric, NMPSO, ISDE+, PiCEA-g, and MaOEA-IBP algorithms demonstrate excellent performance, but RVEA, AnD, NSGAII-SDR, and EFR-RR algorithms demonstrate poor performance. The diversity assessment results obtained based on the PD metric illustrate that MOEAD-URAW, AnD, Pi-MOEA, and RPD-NSGAII algorithms achieve better diversity compared to that achieved by the PiCEA-g, RVEA, EFR-RR, and NSGAIII algorithms. Therefore, the ISDE+, MaOEA-IBP, MOEAD-URAW, and Pi-MOEA algorithms perform consistently better, and the RVEA algorithm performs poorly with respect to the four performance metrics.
- ***VVM problem***: The performance metric results of 15 algorithms on the VVM problem are listed in Table 3. Based on HV and IGD metrics, the MOEAD-URAW algorithm achieves better performance than the rest of the algorithms. Among the remaining algorithms, MaOEA-IBP, PiCEA-g, and AnD perform better than other MOEAs in terms of HV, and the Pi-MOEA,

TABLE 1. Mean and standard deviation values of performance metrics on CSID problem.

CSID Problem				
Algorithms	HV	IGD	GD	PD
AnD	4.821e-1 (8.41e-3)	6.160e-1 (4.04e-2)	7.252e-2 (1.57e-2)	**1.470e+7 (9.07e+5)**
EFR-RR	4.686e-1 (3.60e-3)	8.339e-1 (4.53e-2)	1.098e-1 (2.18e-2)	9.362e+6 (1.43e+6)
ISDE+	5.052e-1 (2.52e-3)	5.608e-1 (3.16e-2)	4.245e-3 (2.62e-3)	1.147e+7 (6.49e+5)
MaOEA-IBP	**5.076e-1 (2.03e-3)**	**4.894e-1 (1.17e-2)**	7.651e-3 (9.24e-3)	1.254e+7 (7.05e+5)
MOEAD-URAW	4.690e-1 (1.08e-2)	5.799e-1 (5.95e-2)	1.712e-2 (6.22e-3)	1.376e+7 (1.28e+6)
NSGAIII	4.519e-1 (1.43e-2)	8.257e-1 (1.43e-1)	2.394e-2 (9.71e-3)	1.245e+7 (1.03e+6)
NSGAII-SDR	3.292e-1 (1.61e-2)	2.146e+0 (2.51e-1)	4.352e-3 (4.53e-3)	1.270e+7 (8.76e+5)
Pi-MOEA	4.626e-1 (5.43e-3)	6.685e-1 (3.02e-2)	1.357e-1 (2.33e-2)	1.287e+7 (1.04e+6)
PiCEA-g	4.704e-1 (5.20e-3)	2.424e+0 (3.28e-1)	3.074e-3 (1.25e-3)	2.491e+6 (6.97e+5)
RPD-NSGAII	4.672e-1 (4.73e-3)	9.052e-1 (5.07e-2)	1.001e-1 (3.52e-2)	1.018e+7 (1.27e+6)
RVEA	4.272e-1 (1.03e-2)	9.354e-1 (4.59e-2)	1.129e-1 (5.63e-2)	8.941e+6 (8.53e+5)
SPEA-R	4.517e-1 (6.03e-3)	9.773e-1 (4.35e-2)	9.243e-2 (2.06e-2)	9.897e+6 (1.09e+6)
t-DEA	4.533e-1 (3.50e-2)	8.884e-1 (3.31e-1)	6.679e-2 (3.52e-2)	8.593e+6 (1.21e+6)
NMPSO	5.056e-1 (2.89e-3)	5.566e-1 (2.71e-2)	**1.798e-3 (1.04e-3)**	1.018e+7 (8.04e+5)
MOFA	4.201e-1 (1.22e-2)	6.269e-1 (6.83e-2)	1.937e-2 (8.33e-3)	1.376e+7 (7.60e+5)

TABLE 2. Mean and standard deviation values of performance metrics on CMD problem.

CMD problem				
Algorithms	HV	IGD	GD	PD
AnD	5.035e-1 (8.64e-3)	3.523e+2 (3.50e+1)	9.835e+2 (1.72e+2)	3.729e+9 (5.18e+8)
EFR-RR	5.016e-1 (6.78e-3)	3.344e+2 (2.87e+1)	2.435e+2 (1.69e+2)	2.344e+9 (5.85e+8)
ISDE+	**5.513e-1 (9.15e-4)**	1.305e+2 (4.48e+0)	6.807e-1 (8.43e-1)	2.506e+9 (1.55e+8)
MaOEA-IBP	5.302e-1 (2.27e-3)	2.036e+2 (2.61e+1)	2.882e+0 (3.07e+0)	2.551e+9 (2.28e+8)
MOEAD-URAW	5.466e-1 (1.34e-3)	**1.046e+2 (3.10e+0)**	3.151e+0 (1.06e+0)	**3.796e+9 (4.55e+8)**
NSGAIII	4.889e-1 (4.72e-2)	5.499e+2 (4.31e+2)	7.993e+0 (3.02e+0)	2.360e+9 (7.32e+8)
NSGAII-SDR	4.979e-1 (6.52e-2)	5.751e+2 (8.19e+2)	9.391e+2 (3.76e+2)	3.161e+9 (1.04e+9)
Pi-MOEA	5.205e-1 (5.86e-3)	1.848e+2 (1.32e+1)	1.346e+1 (5.45e+0)	3.466e+9 (8.38e+8)
PiCEA-g	4.061e-1 (2.34e-2)	1.355e+3 (2.65e+2)	1.854e+0 (1.22e+0)	1.592e+9 (2.41e+8)
RPD-NSGAII	5.107e-1 (3.72e-3)	2.972e+2 (1.80e+1)	1.391e+2 (6.26e+1)	3.403e+9 (6.66e+8)
RVEA	4.878e-1 (3.85e-3)	4.759e+2 (2.97e+1)	1.126e+3 (8.79e+2)	2.207e+9 (6.45e+8)
SPEA-R	4.996e-1 (1.01e-2)	2.924e+2 (3.38e+1)	9.676e+1 (5.33e+1)	2.912e+9 (5.94e+8)
t-DEA	5.095e-1 (6.90e-3)	2.994e+2 (2.86e+1)	2.054e+1 (1.60e+1)	2.369e+9 (3.67e+8)
NMPSO	5.374e-1 (1.31e-2)	2.788e+2 (1.39e+2)	**3.738e-1 (2.43e-1)**	2.386e+9 (1.60e+8)
MOFA	5.170e-1 (1.13e-2)	2.295e+2 (4.71e+1)	5.552e+1 (2.76e+1)	3.347e+9 (5.33e+8)

SPEA-R, and EFR-RR algorithms achieve better performance than other MOEAs in terms of the IGD metric. By contrast, NSGAII-SDR, RVEA, and t-DEA algorithms exhibit poor performance on the VVM problem in terms of HV and IGD. Based on the GD metric, the PiCEA-g, ISDE+, MaOEA-IBP, and MOEAD-URAW algorithms exhibit optimal convergence abilities, and NSGAII-SDR, RPD-NSGAII, AnD, as well as EFR-RR algorithms exhibit less-than-optimal convergence performance on the VVM problem. In terms of the PD metric, NSGAII-SDR, Pi-MOEA, AnD, and MOEAD-URAW algorithms demonstrate better diversity performance than PiCEA-g, RVEA, ISDE+, and t-DEA algorithms. In summary, the MOEAD-URAW performs better compared to other MOEAs consistently on all performance metrics. By contrast, RVEA and t-DEA algorithms exhibit worse performance compared to other algorithms in terms of all performance metrics.

- ***LPMS problem***: The mean and standard deviation values of HV, IGD, GD, and PD metrics obtained by 15 algorithms are listed in Table 4. The ISDE+, MaOEA-IBP, MOFA, and MOEAD-URAW algorithms perform better than RVEA, RPD-NSGAII, SPEA-R, and

TABLE 3. Mean and standard deviation values of performance metrics on VVM problem.

VVM Problem				
Algorithms	HV	IGD	GD	PD
AnD	4.362e-1 (2.70e-3)	3.636e-2 (5.27e-3)	2.813e-3 (2.39e-3)	2.936e+6 (3.40e+5)
EFR-RR	4.343e-1 (3.04e-3)	3.139e-2 (3.98e-3)	2.290e-3 (5.43e-4)	2.771e+6 (2.52e+5)
ISDE+	4.327e-1 (2.38e-3)	5.251e-2 (1.31e-2)	4.130e-5 (3.93e-5)	1.325e+6 (1.21e+5)
MaOEA-IBP	**4.403e-1 (1.76e-3)**	**4.789e-2 (1.24e-2)**	6.576e-5 (5.49e-5)	1.437e+6 (1.65e+5)
MOEAD-URAW	4.405e-1 (1.38e-3)	2.099e-2 (8.24e-4)	3.120e-4 (1.05e-4)	2.787e+6 (2.52e+5)
NSGAIII	4.355e-1 (2.33e-3)	5.134e-2 (1.06e-2)	4.269e-4 (7.36e-5)	1.773e+6 (2.04e+5)
NSGAII-SDR	3.541e-1 (1.75e-2)	1.732e-1 (7.04e-2)	4.596e-2 (3.61e-2)	**5.445e+6 (8.37e+5)**
Pi-MOEA	4.357e-1 (2.46e-3)	2.767e-2 (1.48e-3)	1.440e-3 (2.76e-4)	4.103e+6 (3.39e+5)
PiCEA-g	4.372e-1 (7.32e-4)	8.069e-2 (1.16e-2)	**2.656e-5 (1.19e-5)**	5.855e+5 (7.81e+4)
RPD-NSGAII	4.186e-1 (1.09e-2)	3.771e-2 (4.76e-3)	6.638e-3 (6.37e-3)	2.209e+6 (2.42e+5)
RVEA	4.229e-1 (5.50e-3)	6.945e-2 (1.28e-2)	1.398e-3 (6.97e-4)	6.855e+5 (1.79e+5)
SPEA-R	4.359e-1 (2.01e-3)	3.149e-2 (2.98e-3)	1.435e-3 (3.00e-4)	2.555e+6 (2.98e+5)
t-DEA	4.309e-1 (1.25e-3)	5.324e-2 (1.35e-2)	5.105e-4 (1.36e-4)	1.424e+6 (1.93e+5)
NMPSO	4.112e-1 (1.36e-3)	3.985e-2 (1.68e-3)	5.385e-5 (5.42e-5)	1.029e+6 (1.10e+5)
MOFA	4.023e-1 (1.45e-3)	4.476e-2 (1.68e-3)	4.662e-4 (4.22e-5)	1.461e+6 (1.20e+5)

TABLE 4. Mean and standard deviation values of performance metrics on LPMS problem.

LPMS problem				
Algorithms	HV	IGD	GD	PD
AnD	9.196e-2 (1.04e-3)	8.022e-1 (2.63e-2)	8.104e-2 (1.39e-2)	3.370e+8 (1.24e+7)
EFR-RR	7.156e-2 (7.56e-3)	1.721e+0 (1.81e-1)	1.135e-1 (4.23e-2)	1.965e+8 (1.91e+7)
ISDE+	**9.800e-2 (8.78e-4)**	7.564e-1 (2.66e-2)	3.882e-3 (3.37e-3)	3.367e+8 (1.47e+7)
MaOEA-IBP	9.751e-2 (9.27e-4)	7.014e-1 (1.75e-2)	4.096e-3 (3.66e-3)	3.501e+8 (9.97e+6)
MOEAD-URAW	9.479e-2 (6.08e-4)	**6.692e-1 (9.18e-3)**	1.878e-2 (9.26e-3)	3.743e+8 (8.15e+6)
NSGAIII	8.548e-2 (5.20e-3)	1.394e+0 (1.69e-1)	1.186e-2 (5.29e-3)	2.177e+8 (2.70e+7)
NSGAII-SDR	9.168e-2 (1.24e-3)	8.566e-1 (3.47e-2)	8.718e-2 (1.79e-2)	3.205e+8 (1.51e+7)
Pi-MOEA	8.614e-2 (2.13e-3)	9.409e-1 (3.96e-2)	6.761e-2 (2.98e-2)	3.134e+8 (2.16e+7)
PiCEA-g	9.256e-2 (1.77e-3)	1.743e+0 (3.02e-1)	**2.099e-3 (2.61e-3)**	2.033e+8 (1.76e+7)
RPD-NSGAII	6.503e-2 (5.57e-3)	1.839e+0 (1.71e-1)	6.136e-2 (3.39e-2)	1.515e+8 (2.19e+7)
RVEA	5.654e-2 (3.14e-3)	2.310e+0 (1.39e-1)	1.955e-1 (7.55e-2)	1.060e+8 (1.37e+7)
SPEA-R	6.964e-2 (4.82e-3)	1.830e+0 (1.59e-1)	2.544e-2 (9.06e-3)	1.323e+8 (1.76e+7)
t-DEA	7.322e-2 (5.21e-3)	1.751e+0 (1.95e-1)	1.512e-2 (6.58e-3)	1.485e+8 (2.15e+7)
NMPSO	9.371e-2 (1.46e-3)	8.475e-1 (6.11e-2)	1.164e-2 (6.63e-3)	3.120e+8 (1.82e+7)
MOFA	9.647e-2 (8.09e-4)	6.860e-1 (2.20e-2)	6.114e-2 (1.40e-2)	**3.986e+8 (1.43e+7)**

EFR-RR algorithms in terms of the HV metric. However, in terms of IGD, the MOEAD-URAW, MOFA, MaOEA-IBP, and ISDE+ algorithms perform better than the RVEA, RPD-NSGAII, SPEA-R, and t-DEA algorithms. In terms of GD, PiCEA-g, ISDE+, MaOEA-IBP, and NMPSO algorithms exhibit good convergence; by contrast, RVEA, EFR-RR, NSGAII-SDR, and AnD algorithms exhibit poor convergence. Regarding PD, MOFA, MOEAD-URAW, MaOEA-IBP, and AnD algorithms exhibit better diversity performance than RVEA, SPEA-R, t-DEA, and RPD-NSGAII algorithms. To sum up the performance of MOEAs on the LPMS problem, MOEAD-URAW, MaOEA-IBP, MOFA, and ISDE+ algorithms perform consistently better than RVEA, RPD-NSGAII, SPEA-R, and t-DEA algorithms according to HV, IGD, GD, and PD metrics.

- ***Machining Problem***: The mean and standard deviation values for performance metrics obtained by 15 algorithms on the Machining problem are listed in Table 5. The NMPSO, MaOEA-IBP, ISDE+, and MOEAD-URAW algorithms achieve better performance than PiCEA-g, RPD-NSGAII, SPEA-R, and

TABLE 5. Mean and standard deviation values of performance metrics on machining problem.

Machining Problem				
Algorithms	HV	IGD	GD	PD
AnD	2.162e-1 (8.30e-4)	1.936e-1 (2.90e-3)	4.193e-18 (4.79e-18)	1.136e+8 (2.78e+6)
EFR-RR	1.571e-1 (5.81e-3)	4.046e-1 (2.26e-2)	4.838e-19 (2.65e-18)	5.387e+7 (4.25e+6)
ISDE+	2.182e-1 (9.17e-4)	2.124e-1 (6.11e-3)	0.000e+0 (0.00e+0)	9.985e+7 (2.81e+6)
MaOEA-IBP	2.187e-1 (9.52e-4)	1.975e-1 (5.12e-3)	9.080e-18 (2.44e-18)	1.108e+8 (3.21e+6)
MOEAD-URAW	2.164e-1 (7.63e-4)	**1.814e-1 (1.92e-3)**	1.175e-19 (6.43e-19)	**1.229e+8 (4.82e+6)**
NSGAIII	2.021e-1 (2.13e-3)	2.457e-1 (7.87e-3)	3.323e-19 (1.82e-18)	7.804e+7 (4.25e+6)
NSGAII-SDR	2.123e-1 (1.93e-3)	2.435e-1 (1.35e-2)	0.000e+0 (0.00e+0)	9.546e+7 (5.91e+6)
Pi-MOEA	1.985e-1 (2.24e-3)	2.496e-1 (8.14e-3)	5.673e-19 (2.19e-18)	7.965e+7 (3.73e+6)
PiCEA-g	7.051e-2 (1.50e-2)	9.624e-1 (1.31e-1)	0.000e+0 (0.00e+0)	1.009e+7 (3.85e+6)
RPD-NSGAII	1.642e-1 (7.19e-3)	3.610e-1 (3.88e-2)	1.112e-17 (6.59e-18)	5.284e+7 (4.25e+6)
RVEA	1.539e-1 (5.41e-3)	4.375e-1 (2.18e-2)	0.000e+0 (0.00e+0)	3.937e+7 (3.70e+6)
SPEA-R	1.563e-1 (5.03e-3)	3.909e-1 (1.40e-2)	0.000e+0 (0.00e+0)	4.531e+7 (3.05e+6)
t-DEA	1.738e-1 (3.65e-3)	3.419e-1 (9.51e-3)	0.000e+0 (0.00e+0)	4.867e+7 (4.29e+6)
NMPSO	**2.190e-1 (1.13e-3)**	2.194e-1 (8.17e-3)	1.507e-18 (3.11e-18)	9.882e+7 (3.08e+6)
MOFA	2.145e-1 (9.73e-4)	1.897e-1 (4.23e-3)	0.000e+0 (0.00e+0)	1.139e+8 (3.87e+6)

TABLE 6. Mean and standard deviation values of performance metrics on WRP problem.

WRP Problem				
Algorithms	HV	IGD	GD	PD
AnD	7.771e-1 (2.56e-3)	1.022e+5 (1.83e+4)	1.050e+4 (6.09e+3)	1.348e+13 (1.42e+12)
EFR-RR	7.565e-1 (4.42e-3)	1.392e+5 (1.64e+4)	1.523e+4 (7.57e+3)	9.445e+12 (2.00e+12)
ISDE+	**7.874e-1 (3.56e-3)**	6.680e+4 (8.51e+3)	1.312e+2 (1.41e+2)	7.620e+12 (5.59e+11)
MaOEA-IBP	7.842e-1 (5.84e-3)	7.315e+4 (7.13e+3)	1.437e+2 (1.20e+2)	5.361e+12 (1.35e+12)
MOEAD-URAW	7.851e-1 (3.13e-3)	**4.956e+4 (2.84e+3)**	3.103e+3 (4.42e+3)	1.056e+13 (1.07e+12)
NSGAIII	7.715e-1 (4.40e-3)	8.300e+4 (1.12e+4)	2.371e+3 (4.96e+2)	9.266e+12 (1.98e+12)
NSGAII-SDR	7.226e-1 (1.64e-2)	1.434e+5 (5.60e+4)	3.304e+2 (1.77e+2)	7.276e+12 (1.83e+12)
Pi-MOEA	7.568e-1 (5.01e-3)	1.299e+5 (1.26e+4)	4.352e+4 (1.13e+4)	**1.883e+13 (2.65e+12)**
PiCEA-g	7.681e-1 (5.45e-3)	9.209e+4 (1.55e+4)	2.088e+2 (1.19e+2)	5.529e+12 (7.10e+11)
RPD-NSGAII	7.448e-1 (6.83e-3)	1.656e+5 (2.58e+4)	1.519e+4 (1.06e+4)	8.650e+12 (1.61e+12)
RVEA	7.031e-1 (1.57e-2)	2.211e+5 (1.69e+4)	8.516e+4 (4.49e+4)	4.379e+12 (1.19e+12)
SPEA-R	7.404e-1 (4.14e-3)	1.642e+5 (1.37e+4)	2.058e+4 (1.12e+4)	1.020e+13 (1.26e+12)
t-DEA	7.518e-1 (6.60e-3)	2.584e+5 (6.56e+4)	2.627e+3 (1.19e+3)	6.890e+12 (1.32e+12)
NMPSO	7.654e-1 (1.25e-2)	9.061e+4 (2.16e+4)	**5.843e+1 (1.04e+2)**	3.549e+12 (3.88e+11)
MOFA	7.644e-1 (7.48e-3)	7.138e+4 (5.68e+3)	5.675e+3 (1.36e+3)	6.477e+12 (1.02e+12)

RVEA algorithms in terms of HV. The MOEAD-URAW, MOFA, AnD, and MaOEA-IBP algorithms achieve better performance than PiCEA-g, RPD-NSGAII, SPEA-R, and RVEA algorithms in terms of IGD. However, in terms of GD, the performance of all metrics is similar/at par in solving Machining Problem. In terms of PD, MOEAD-URAW, MOFA, AnD, and MaOEA-IBP algorithms present excellent performance compared with PiCEA-g, RPD-NSGAII, SPEA-R, and RVEA algorithms. In summary, MOEAD-URAW, MOFA, AnD, and MaOEA-IBP algorithms consistently perform better than PiCEA-g, RPD-NSGAII, SPEA-R, and RVEA in terms of all performance metrics.

- ***WRP problem***: The mean and standard deviation values of HV, IGD, GD, and PD metrics obtained by the 15 algorithms are listed in Table 6. The ISDE+, MOEAD-URAW, MaOEA-IBP, and AnD algorithms exhibit better performance, and RVEA, NSGAII-SDR, SPEA-R, and RPD-NSGAII algorithms perform poorly with respect to the HV metric. However, in terms of IGD, MOEAD-URAW, ISDE+, MOFA, and MaOEA-IBP algorithms exhibit good performance, and t-DEA, RVEA, SPEA-R, and RPD-NSGAII algorithms exhibit

TABLE 7. Mean and standard deviation values of performance metrics on WRCD problem.

WRCD problem				
Algorithms	HV	IGD	GD	PD
AnD	1.444e-1 (8.63e-4)	**6.874e+1 (1.94e+0)**	2.087e+0 (3.99e-1)	**1.406e+11 (5.00e+9)**
EFR-RR	9.347e-2 (4.37e-3)	1.357e+2 (5.69e+0)	3.621e+0 (5.58e-1)	7.048e+10 (6.00e+9)
ISDE+	1.418e-1 (3.03e-3)	2.036e+2 (4.57e+1)	**6.444e-3 (2.24e-2)**	7.851e+10 (6.39e+9)
MaOEA-IBP	1.444e-1 (1.14e-3)	1.197e+2 (2.08e+1)	1.534e-1 (1.04e-1)	9.733e+10 (9.35e+9)
MOEAD-URAW	1.351e-1 (2.17e-3)	7.255e+1 (2.83e+0)	7.571e-1 (4.15e-1)	1.327e+11 (7.69e+9)
NSGAIII	1.324e-1 (2.65e-3)	1.257e+2 (2.30e+1)	1.051e+0 (4.60e-1)	8.117e+10 (7.45e+9)
NSGAII-SDR	1.390e-1 (2.18e-3)	7.951e+1 (4.49e+0)	7.429e-1 (2.90e-1)	1.211e+11 (8.84e+9)
Pi-MOEA	1.261e-1 (3.29e-3)	8.279e+1 (2.89e+0)	2.877e+0 (8.79e-1)	1.120e+11 (9.96e+9)
PiCEA-g	3.287e-2 (8.94e-3)	3.634e+2 (3.73e+1)	8.874e-3 (2.36e-2)	2.301e+10 (5.32e+9)
RPD-NSGAII	1.057e-1 (1.30e-3)	1.273e+2 (4.38e+0)	2.127e+0 (3.90e-1)	6.274e+10 (4.72e+9)
RVEA	6.977e-2 (3.08e-3)	1.834e+2 (1.27e+1)	1.778e+0 (8.90e-1)	3.720e+10 (3.85e+9)
SPEA-R	9.024e-2 (2.73e-3)	1.428e+2 (7.00e+0)	3.609e+0 (2.91e-1)	6.250e+10 (4.48e+9)
t-DEA	1.059e-1 (2.69e-3)	1.324e+2 (7.93e+0)	1.240e+0 (4.13e-1)	5.573e+10 (4.20e+9)
NMPSO	**1.486e-1 (1.00e-3)**	1.115e+2 (3.09e+1)	1.619e-1 (8.92e-2)	1.037e+11 (6.25e+9)
MOFA	1.226e-1 (5.20e-3)	7.111e+1 (2.43e+0)	1.382e+0 (1.52e-1)	1.285e+11 (4.50e+9)

TABLE 8. Mean and standard deviation values of performance metrics on CCD problem.

CCD problem				
Algorithms	HV	IGD	GD	PD
AnD	5.949e-2 (2.78e-3)	1.624e+0 (8.52e-2)	1.664e-2 (2.53e-3)	**3.022e+10 (2.14e+9)**
EFR-RR	5.643e-2 (3.54e-3)	2.298e+0 (2.71e-1)	1.212e-2 (2.00e-3)	1.866e+10 (1.47e+9)
ISDE+	**6.067e-2 (2.75e-3)**	1.757e+0 (1.69e-1)	1.300e-2 (6.18e-3)	2.790e+10 (1.85e+9)
MaOEA-IBP	5.663e-2 (3.76e-3)	1.929e+0 (3.64e-1)	1.441e-2 (3.01e-3)	2.799e+10 (1.73e+9)
MOEAD-URAW	3.904e-2 (4.14e-3)	2.059e+0 (2.41e-1)	1.804e-2 (4.33e-3)	1.972e+10 (1.98e+9)
NSGAIII	5.171e-2 (4.50e-3)	1.700e+0 (1.68e-1)	1.861e-2 (7.53e-3)	2.680e+10 (2.03e+9)
NSGAII-SDR	1.778e-3 (4.44e-4)	1.555e+1 (1.22e+0)	3.028e-2 (5.78e-3)	5.144e+9 (8.86e+8)
Pi-MOEA	5.534e-2 (3.16e-3)	**1.526e+0 (9.03e-2)**	1.487e-2 (2.28e-3)	2.727e+10 (1.23e+9)
PiCEA-g	5.971e-2 (3.14e-3)	2.321e+0 (2.02e-1)	**8.465e-3 (2.13e-3)**	1.563e+10 (1.25e+9)
RPD-NSGAII	5.800e-2 (3.83e-3)	1.888e+0 (1.57e-1)	1.249e-2 (1.89e-3)	2.144e+10 (1.58e +9)
RVEA	8.161e-3 (3.37e-3)	9.933e+0 (2.05e+0)	5.021e-2 (2.26e-2)	4.963e+9 (1.05e+9)
SPEA-R	5.005e-2 (3.80e-3)	1.816e+0 (8.32e-2)	1.355e-2 (2.81e-3)	2.038e+10 (1.73e+9)
t-DEA	5.913e-2 (4.35e-3)	2.217e+0 (3.69e-1)	9.925e-3 (1.35e-3)	1.918e+10 (1.57e+9)
NMPSO	6.007e-2 (4.41e-3)	1.668e+0 (1.81e-1)	1.121e-2 (2.58e-3)	2.715e+10 (2.11e+9)
MOFA	3.233e-2 (3.48e-3)	2.195e+0 (3.21e-1)	1.892e-2 (2.23e-3)	1.551e+10 (1.22e+9)

poor performance. In terms of GD, NMPSO, ISDE+, MaOEA-IBP, and PiCEA-g algorithms show a good convergence, but RVEA, Pi-MOEA, SPEA-R, and EFR-RR algorithms fail to converge on the WRP problem. Regarding PD, Pi-MOEA, AnD, MOEAD-URAW, and SPEA-R algorithms exhibit good diversity capabilities, but NMPSO, RVEA, MaOEA-IBP, PiCEA-g, and t-DEA algorithms exhibit poor diversity capabilities. To sum up the performance of MOEAs in solving the WRP problem, the MOEAD-URAW algorithm performs better consistently than RVEA, RPD-NSGAII, and t-DEA algorithms with respect to the four performance metrics worse.

- ***WRCD problem***: The performance metric values obtained by MOEAs on the WRCD problem are listed in Table 7. The NMPSO, AnD, MaOEA-IBP, and ISDE+ algorithms perform better, and the PiCEA-g, RVEA, SPEA-R, and EFR-RR perform worse according to the HV. Furthermore, in terms of IGD, AnD, MOFA, MOEAD-URAW, and NSGAII-SDR, algorithms exhibit better performance, and the PiCEA-g, ISDE+, RVEA, and SPEA-R algorithms perform poorly. Regarding

TABLE 9. Mean and standard deviation values of performance metrics on RWD problem.

RWD problem				
Algorithms	HV	IGD	GD	PD
AnD	**7.209e-3 (2.57e-4)**	9.231e+2 (9.76e+1)	5.442e+0 (4.72e-1)	7.191e+12 (3.58e+11)
EFR-RR	3.159e-3 (3.82e-4)	1.397e+3 (2.59e+2)	7.260e+0 (3.10e+0)	1.935e+12 (4.05e+11)
ISDE+	8.755e-3 (2.36e-4)	9.051e+2 (1.55e+2)	2.262e+0 (5.18e-1)	**6.231e+12 (4.77e+11)**
MaOEA-IBP	6.824e-3 (2.45e-4)	6.627e+2 (8.47e+1)	3.782e+0 (5.85e-1)	5.886e+12 (3.74e+11)
MOEAD-URAW	3.215e-3 (3.47e-4)	5.691e+2 (4.78e+1)	7.536e+0 (1.29e+0)	6.342e+12 (4.49e+11)
NSGAIII	1.989e-3 (2.84e-4)	2.187e+3 (2.05e+2)	4.189e+0 (5.20e-1)	3.047e+12 (3.16e+11)
NSGAII-SDR	1.014e-3 (1.48e-4)	3.013e+3 (1.27e+2)	8.726e+0 (1.05e+0)	4.434e+12 (2.53e+11)
Pi-MOEA	5.170e-3 (2.90e-4)	**5.506e+2 (2.50e+1)**	8.934e+0 (8.51e-1)	5.975e+12 (3.39e+11)
PiCEA-g	2.450e-3 (3.88e-4)	2.571e+3 (2.32e+2)	2.506e+0 (2.28e+0)	3.190e+11 (1.45e+11)
RPD-NSGAII	**3.439e-3 (6.40e-4)**	9.259e+2 (1.80e+2)	2.128e+1 (1.06e+1)	2.711e+12 (4.25e+11)
RVEA	6.927e-4 (1.31e-4)	3.246e+3 (1.93e+2)	3.072e+1 (1.53e+1)	1.035e+12 (1.42e+11)
SPEA-R	2.992e-3 (5.11e-4)	1.244e+3 (1.56e+2)	2.059e+1 (5.44e+0)	2.176e+12 (3.13e+11)
t-DEA	4.472e-3 (4.91e-4)	1.395e+3 (2.19e+2)	6.942e+0 (2.82e+0)	2.309e+12 (4.38e+11)
NMPSO	8.172e-4 (2.31e-4)	1.035e+3 (1.04e+2)	4.017e+1 (1.11e+1)	1.702e+12 (3.20e+11)
MOFA	2.201e-3 (2.44e-4)	9.855e+2 (1.37e+2)	1.503e+1 (1.10e+0)	4.052e+12 (3.30e+11)

TABLE 10. Mean and standard deviation values of performance metrics on GAA design problem.

GAA design problem				
Algorithms	HV	IGD	GD	PD
AnD	5.335e-2 (1.98e-3)	1.374e+2 (1.30e+1)	1.087e+0 (8.36e-2)	5.203e+12 (1.85e+11)
EFR-RR	2.152e-2 (2.31e-3)	6.334e+2 (9.27e+1)	3.419e-1 (1.16e-1)	9.735e+11 (1.75e+11)
ISDE+	6.416e-2 (3.59e-3)	1.560e+2 (2.76e+1)	3.213e-1 (1.01e-1)	4.206e+12 (4.25e+11)
MaOEA-IBP	6.134e-2 (2.26e-3)	1.351e+2 (2.96e+1)	3.970e-1 (8.58e-2)	3.592e+12 (2.45e+11)
MOEAD-URAW	2.774e-2 (3.04e-3)	**1.248e+2 (9.85e+0)**	9.008e-1 (1.85e-1)	3.755e+12 (3.22e+11)
NSGAIII	2.571e-2 (1.97e-3)	2.168e+2 (5.06e+1)	1.392e+0 (2.20e-1)	3.409e+12 (2.46e+11)
NSGAII-SDR	1.214e-2 (9.07e-4)	9.074e+2 (1.07e+2)	5.510e-1 (6.87e-2)	1.386e+12 (9.41e+10)
Pi-MOEA	4.041e-2 (2.14e-3)	1.355e+2 (9.81e+0)	3.157e+0 (3.45e-1)	4.920e+12 (2.34e+11)
PiCEA-g	6.738e-3 (7.44e-4)	4.460e+2 (4.70e+1)	1.678e+0 (1.30e-1)	4.241e+12 (1.37e+11)
RPD-NSGAII	3.217e-2 (3.08e-3)	2.883e+2 (4.05e+1)	9.322e-1 (2.55e-1)	1.629e+12 (1.44e+11)
RVEA	1.091e-2 (1.78e-3)	1.127e+3 (1.48e+2)	1.251e+0 (6.59e-1)	2.818e+11 (5.18e+10)
SPEA-R	1.708e-2 (1.64e-3)	2.152e+2 (1.95e+1)	1.877e+0 (2.22e-1)	1.603e+12 (1.13e+11)
t-DEA	3.142e-2 (2.66e-3)	3.998e+2 (4.23e+1)	5.027e-1 (8.10e-2)	1.469e+12 (1.46e+11)
NMPSO	**7.379e-2 (2.99e-3)**	1.271e+2 (1.28e+1)	**2.204e-1 (1.31e-1)**	**6.556e+12 (1.54e+11)**
MOFA	9.560e-3 (1.33e-3)	2.031e+2 (2.09e+1)	2.249e+0 (1.04e-1)	5.119e+12 (1.67e+11)

GD, ISDE+, PiCEA-g, MaOEA-IBP, and NMPSO algorithms attain good convergence, and by contrast, the algorithms EFR-RR, Pi-MOEA, SPEA-R, RPD-NSGAII exhibit bad convergence performance. With regard to PD, AnD, MOEAD-URAW, MOFA, and NSGAII-SDR exhibit good diversity, and PiCEA-g, RVEA, t-DEA, and SPEA-R algorithms exhibit poor diversity performance. In summary, AnD and MOEAD-URAW algorithms consistently have better than other MOEAs, and the SPEA-R and RPD-NSGAII algorithms perform worse than other MOEAs in terms of all the performance metrics.

- ***CCD problem***: The mean and standard deviation values of HV, IGD, GD, and PD obtained by 15 algorithms on the CCD problem are listed in Table 8. Algorithms ISDE+, NMPSO, PiCEA-g, and AnD attain better performance than other MOEAs, and NSGAII-SDR, RVEA, MOFA, and MOEAD-URAW, algorithms perform poorly in terms of HV metric compared to other MOEAs. Regarding IGD, Pi-MOEA, AnD, NMPSO, and NSGAIII, algorithms demonstrate better performance than the NSGAII-SDR, RVEA, PiCEA-g, and EFR-RR algorithms. However, in terms of convergence performance, the PiCEA-g, t-DEA, NMPSO, and EFR-RR algorithms exhibit excellent performance, and the algorithms RVEA, NSGAII-SDR, MOFA, and NSGAIII exhibit poor performance. In terms of PD, AnD, MaOEA-IBP, ISDE+, and Pi-MOEA exhibit good diversity performance compared to that exhibited by RVEA, NSGAII-SDR, MOFA, and PiCEA-g algorithms. In summary, the ISDE+, NMPSO, AnD algorithm has consistently better performance than other MOEAs in terms of performance metrics, and MOFA, NSGAII-SDR, and RVEA algorithms perform poorly with respect to the performance metrics.

- ***RWD problem***: The performance metric values obtained by algorithms on the RWD problem are listed in Table 9. The RPD-NSGAII, SPEAR, t-DEA, and PiCEA-g algorithms show a better performance than RVEA, NSGAII-SDR, NMPSO, and ISDE+ algorithms perform inferior according to the HV. In terms of IGD, Pi-MOEA, MOEAD-URAW, MaOEA-IBP, and AnD algorithms exhibit highly efficient performance, whereas EFR-RR, PiCEA-g, t-DEA, and NSGAIII exhibit poor performance. With regard to GD, the AnD, ISDE+, NSGAIII, and MaOEA-IBP algorithms possess good convergence capabilities, but RVEA, NMPSO, RPD-NSGAII, and SPEA-R algorithms have poor convergence. In terms of PD, ISDE+, MOEAD-URAW, t-DEA, and RPD-NSGA algorithms exhibit good diversity performance, and RVEA, PiCEA-g, NSGAII-SDR, and SPEA-R algorithms exhibit poor diversity performance. To sum up the performance of MOEAs on the RWD problem, AnD and RVEA algorithms exhibit the best and worse performance, respectively.
- ***GAA design problem***: The mean and standard deviation of the performance metrics obtained by 15 algorithms on the GAA design problem are listed in Table 10. The NMPSO, ISDE+, MaOEA-IBP, and AnD algorithms perform better than PiCEA-g, MOFA, RVEA, and NSGAII-SDR algorithms in terms of HV metric. Regarding IGD, the MOEAD-URAW, NMPSO, MaOEA-IBP, and Pi-MOEA algorithms exhibit better performance than RVEA, NSGAII-SDR, EFR-RR, and PiCEA-g algorithms. In terms of GD, NMPSO, ISDE+, EFR-RR, and MaOEA-IBP algorithms present good convergence properties and Pi-MOEA, MOFA, SPEA-R and PiCEA-g present poor convergence. With regard to PD, NMPSO, AnD, MOFA, and Pi-MOEA algorithms exhibit good diversity compared to RVEA, EFR-RR, NSGAII-SDR, and t-DEA algorithms. In summary, NMPSO, ISDE+, and MaOEA-IBP algorithms perform consistently better than RVEA and PiCEA-g algorithms with respect to the performance metrics.

C. FURTHER ANALYSIS

1) OVERALL PERFORMANCE OF THE ALGORITHMS

In the preceding sections, a comprehensive analysis was conducted on the efficacy of the various algorithms in addressing each real-world problem. This section presents an analysis of the overall performance of the selected algorithms across ten real-world problems. Furthermore, to assess the effectiveness of 15 algorithms, a Friedman test was performed for hypothesis testing utilizing the KEEL program [61]. Table 11 displays the mean ranks achieved by every algorithm for each performance metric, as per the Friedman test. The p-value and χ^2 value for each performance metric have been presented in our analysis. Table 11 displays the p-value of 0 for each performance metric, suggesting statistically significant distinctions among the algorithms listed. Furthermore, the chi-squared values pertaining to HV, IGD, GD, and PD are 62.0586, 84.5826, 62.4532, and 69.3693, correspondingly.

TABLE 11. Average rank obtained by each algorithm according to friedman test.

Friedman Test Ranking				
Algorithms	HV	IGD	GD	PD
AnD	5.25	4.9	10.3	2.5
EFR-RR	9.4	10.6	10.5	9.65
ISDE+	3.7	6	2.6	6.55
MaOEA-IBP	3.65	4.65	5.3	6.8
MOEAD-URAW	5.1	2.4	7.3	3.5
NSGA-III	8.8	8.6	7.3	8.5
NSGAII-SDR	11.7	10.7	8.5	7.8
Pi-MOEA	7.6	4.85	10.9	4.7
Picea-g	8.7	13.1	3.9	12.8
RPD-NSGAII	8.8	9.9	11.1	8.25
RVEA	13.8	13.3	12.5	13.9
SPEA-R	10.4	9.6	10.2	9.6
t-DEA	8.5	11.3	6.7	10.7
NMPSO	4.8	4.9	4.45	8.4
MOFA	9.8	5.2	8.45	6.35
p-value	**0**	**0**	**0**	**0**
χ^2	**62.0586**	**84.5826**	**62.4532**	**69.3693**

- Table 11 presents a comparative analysis of various multi-objective evolutionary algorithms (MOEAs) based on their performance metrics, namely HV, IGD, GD, and PD, across multiple real-world problems.
- The results indicate that MaOEA-IBP, ISDE+, NMPSO, and MOEAD-URAW algorithms consistently outperform other MOEAs in terms of the aforementioned metrics.
- The findings indicate that ISDE+, MaOEA-IBP, and MOEAD-URAW algorithms exhibit superior performance compared to other MOEAs in addressing most of the problems analyzed, as evidenced by the values in Tables 1 - 10.
- Furthermore, the AnD and Pi-MOEA algorithms exhibit favorable performance with respect to the HV, IGD, and PD metrics, albeit demonstrating comparatively inferior performance concerning the GD metric.
- The aforementioned observation suggests that the algorithms AnD and Pi-MOEA face difficulties in achieving convergence. The algorithms EFR-RR, NSGAII-SDR, NSGAIII, t-DEA, SPEA-R, and RPD-NSGAII demonstrate satisfactory performance in terms of the HV, IGD, GD, and PD metrics.

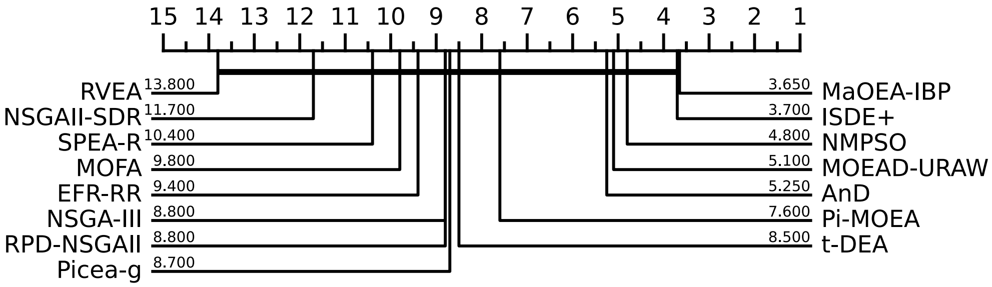

FIGURE 2. **Critical diagram plots based on HV metric.**

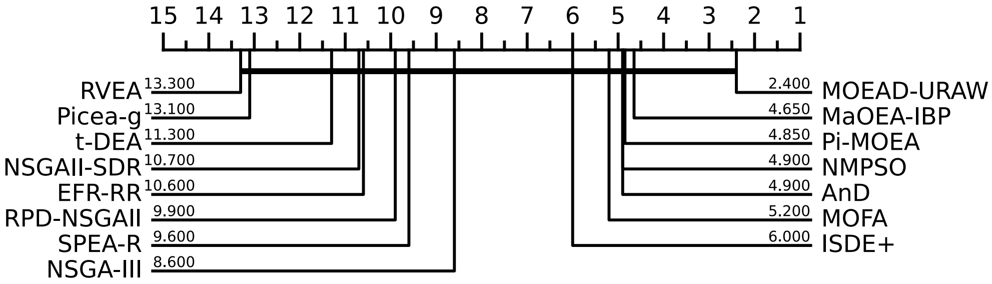

FIGURE 3. **Critical diagram plots based on IGD metric.**

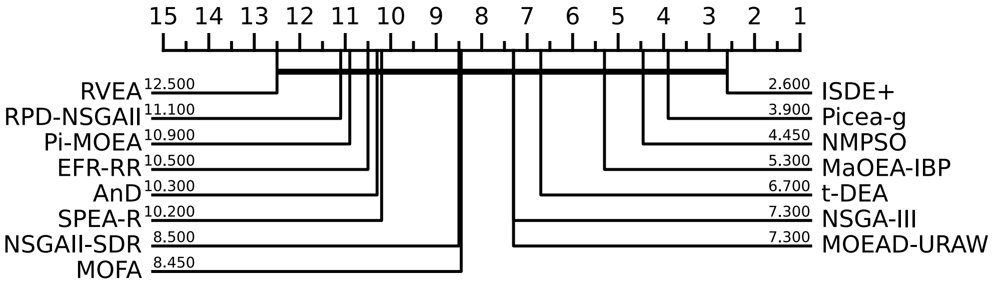

FIGURE 4. **Critical diagram plots based on GD metric.**

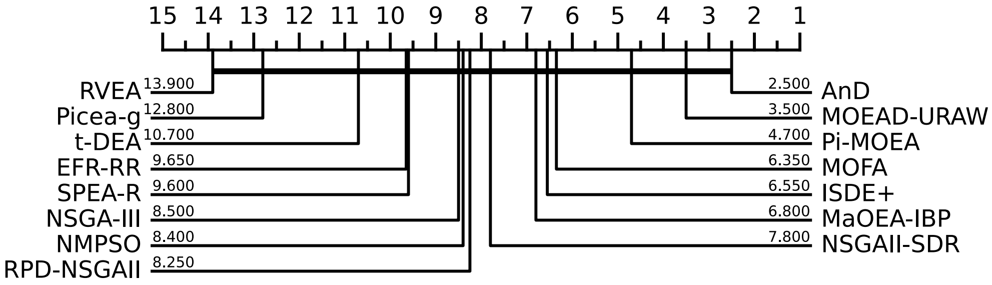

FIGURE 5. **Critical diagram plots based on PD metric.**

- The PiCEA-g and RVEA algorithms demonstrate inadequate performance across all real-world problems in relation to every performance metric. The PiCEA-g algorithm's proficiency in GD is noteworthy but may be deemed premature convergence.

Furthermore, the critical diagram plots are presented in Figs. 2 - 5 for analyzing the algorithms' performance, as they provide detailed information on the subject [12]. The crucial diagram plot displays the algorithms ranked according to their performance indicators. The algorithms with the highest performance indicators are positioned on the right side of the plot, as illustrated in Figures 2 - 5. Conversely, the algorithms with the lowest performance indicators are situated on the left side of the plot. Based on the data presented in Figures 2 - 5, it can be observed that MaOEA-IBP, ISDE+, and MOEAD-URAW consistently exhibit superior performance across all performance metrics, as evidenced by their consistent placement on the right-hand side of the visual representation. The image depicts a consistent pattern wherein the algorithms RVEA, SPEA-R, and EFR-RR consistently exhibit inferior performance across all performance measures, as evidenced by their placement on the left side of the image.

2) DISCUSSIONS AND ANALYSIS

Many algorithms have been suggested in the literature to address many-objective optimization problems, and they have demonstrated exceptional performance across various benchmark problems. The present study conducted a comparative analysis to assess algorithms' performance in real-world scenarios involving many-objective problems. Hence, 15 algorithms have been chosen from the existing literature, with each algorithm being classified into separate categories as described in Section II-C. In other words, as previously stated, several approaches (namely Pareto-dominance-based, indicator-based, decomposition-based, and preference-based) have been suggested to handle MaOPs. The algorithms selected in this study serve as a representation of these methodologies. The most important insights of this comparative study are outlined as follows:

- As stated by the no-free lunch theory, no single algorithm can outperform all other algorithms on the entire set of test problems. A similar conclusion can be drawn from the experimental results on each individual problem, where no algorithm performs better than the remaining algorithms on all ten real-world problems.
- Among the different approaches proposed for handling MaOPs, the indicator-based approaches have demonstrated better performance when compared to the remaining methodologies on the ten real-world many-objective problems. The aforementioned observation can be derived from the critical diagram plots depicted in Figures 2 through 5. These plots indicate that the indicator-based algorithms, namely ISDE+ and MaOEA-IBP, consistently exhibit superior performance, as they are consistently positioned on the right-hand side of the plot. Therefore, indicator-based methodologies exhibit strong performance on real-world problems by employing suitable parameter selection and optimizing population member indicator values.
- Among the two algorithms (EFR-RR and MOEAD-URAW) chosen to represent decomposition-based approaches with weight vector adaptations, MOEAD-URAW performs better on real-world problems, and in contrast, the EFR-RR algorithm fails drastically. The disparity in performance between the two algorithms indicates that decomposition-based methods are susceptible to weight vector adaptation. Therefore, developing efficient methods for generating weight vectors would enhance the functionality of decomposition-based approaches in real-world scenarios.
- In contrast, decomposition-based approaches with reference vector strategies (RVEA, SPEA-R, and RPD-NSGAII) are unable to handle real-world problems effectively. Due to the fact that the Pareto front of real-world problems is unknown, generating appropriate reference vectors is a daunting task. In consequence, these approaches perform poorly on real-world problems.
- The performance of the relaxed dominance approaches, namely NSGAII-SDR and t-DEA, on real-world problems is suboptimal due to their susceptibility to becoming trapped in local optima. By contrast, the methodologies that utilize additional selection metrics, namely Pi-MOEA and AnD, exhibit superior performance when applied to real-world problems. NSGA-III algorithm that incorporates reference points selection criteria as an additional metric exhibits suboptimal performance when applied to real-world problems due to the inherent challenge of approximating reference points in such scenarios.
- The preference-based approaches necessitate external input in order to incorporate preferences to handle MaOPs. In contrast, it is challenging to articulate preferences in real-world scenarios, rendering these approaches (PiCEA-g) incapable of addressing real-world problems.
- Due to their efficient search capabilities, the swarm intelligence approaches (NMPSO and MOFA) achieve respectable results with real-world problems.

3) OPEN ISSUES AND FUTURE DIRECTIONS

1) The availability of real-world many-objective problems in the public domain is limited. There are a variety of factors why only real-world problems are discussed publicly.
    - Corporate companies may sometimes conceal information related to mathematical definitions of real-world problems to safeguard their trade secrets.
    - Occasionally, the reasons are implementation-related, such as when proprietary software is required to conduct the evaluations. For instance, Airfoil designing problem proposed in [62] requires XFOIL [63] software for evaluation.
    - It is also conceivable that individuals do not disclose their problems to the public due to a perceived lack of advantages associated with such an action.

   Consequently, there is a need to develop a test suite for benchmarking real-world problems in order to evaluate the efficacy of algorithms and make them available to the public. In the case of security or other issues, real-world many-objective problems can be formulated as a black-box optimization problem, in which the providers can conceal the problem characteristics and the users can analyze the algorithms.
2) Algorithms specifically developed for addressing many-objective problems exhibit exceptional

performance on synthetic and benchmark problems. Nonetheless, this particular performance does not accurately translate to the effective resolution of numerous many-objective problems encountered in practical applications. Consequently, designing algorithms that exhibit superior performance on real-world problems is imperative.

IV. CONCLUSION AND FUTURE WORKS

In this paper, the search behaviors of 15 state-of-the-art algorithms (AnD, EFR-RR, ISDE+, MaOEA-IBP, MOEAD-URAW, NSGAIII, NSGAII-SDR, Pi-MOEA, PiCEA-g, RPD-NSGAII, RVEA, SPEA-R, t-DEA, NMPSO, and MOFA) in the ten real-world problems from different domains were examined. The algorithms considered in this article belong to different categories, such as Pareto dominance-based (relaxed dominance and additional metrics), indicator-based, decomposition-based (weight vector and reference set), and preference-based approaches. Further, HV, IGD, GD, and PD were used as performance metrics to examine the performance of MOEAs. The experimental results obtained using real-world problems demonstrate that the indicator-based algorithms (ISDE+ and MaOEA-IBP) have consistently outperformed the other algorithms in terms of the HV, IGD, GD, and PD metrics. On the other hand, the Pareto-dominance techniques based on secondary selection criteria (Pi-MOEA and NSGAIII), as well as the decomposition-based algorithm with a weight vector (MOEAD-URAW), have both shown improved performance. The Pareto-dominance techniques with relaxed dominance (t-DEA and NSGAII-SDR) perform inconsistently because they perform better on certain measures while performing worse on others. According to the performance measures, NMPSO among the swarm intelligence algorithms performs better, and MOFA on the other hand, performs quite well. In terms of the performance metrics, the preference-based techniques and the decomposition-based approaches based on reference vectors (RVEA, SPEA-R, and EFR-RR) perform poorly.

In this comparative study, we covered a few many-objective real-world applications to examine the performance of the algorithms. As a part of the future study, we aim to develop a test suite for benchmarking the many-objective applications.

APPENDIX

The mathematical models for the real-world many-objective problems considered in this paper are as follows:

- **Car side impact design problem**

$f_1(x) = 1.98 + 4.9x_1 + 6.67x_2 + 6.98x_3 + 4.01x_4 + 1.78x_5 + 10^{-5}x6 + 2.73x_7$

$f_2(x) = 4.72 - 0.5x_4 - 0.19x_2x_3$

$f3(x) = 0.5\,(VMBP(x) + VFD(x))$

$g_1(x) = 1 - 1.16 + 0.3717x_2x_4 + 0.0092928x_3 \geq 0$

$g_2(x) = 0.32 - 0.261 + 0.0159x_1x_2 + 0.06486x_1 + 0.019 \times 2x7 - 0.0144x_3x_5 - 0.0154464x_6 \geq 0$

$g_3(x) = 0.32 - 0.214 - 0.0082x_5 + 0.04520x_1 + 0.0135168x_1 - 0.03099x_2x + 6 + 0.018x_2x_7 - 0.007176x_3 - 0.0232x_3 + 0.00364x_5x_6 + 0 : 018x_2^2 \geq 0$

$g_4(x) = 0.32 - 0 - 74 + 0.61x_2 + 0.031296x_3 + 0.031872 \times 7 - 0.227x_2^2 \geq 0$

$g_5(x) = 32 - 28.98 - 3.818x_3 + 4.2x_1x_2 - 1.27296x_6 + 2.68065x_7 \geq 0$

$g_6(x) = 32 - 33.86 - 2.95x_3 + 5.057x_1x_2 + 3.795x_2 + 3.4431x_7 - 1.45728 \geq 0$

$g_7(x) = 32 - 46.36 + 9.9x_2 + 4.4505x_1 \geq 0$

$g_8(x) = 4 - f_2(x) \geq 0$

$g_9(x) = 9.9 - VMBP(x) \geq 0$

$g10(x) = 15 : 7 - VFD(x) \geq 0$

$VMBP(x) = 10.58 - 0.674x_1x_2 - 0.67275x_2$

$VFD(x) = 16 : 45 - 0.489x_3x_7 - 0.843x_5x_6$

$f_4(x) = \sum_{i=1}^{10} max\{g_i(x), 0\}$

where $x_1 \in [0.5, 1.5]$, $x_2 \in [0.45, 1.35]$, $x_3 \in [0.5, 1.5]$, $x_4 \in [0.5, 1.5]$, $x_5 \in [0.875, 2.625]$, $x_6 \in [0.4, 1.2]$, $x_7 \in [0.4, 1.2]$

- **Conceptual Marine Design**

$f_1(x) = \frac{annualcosts}{annualcargo}$

$f2(x) = W_s + W_o + W_m;$

$f_3(x) = -Cargo_{DWT} \times RTPA$

$g_1(x) = \frac{L}{B} - 6 \geq 0$

$g_2(x) = 15 - \frac{L}{D} \geq 0$

$g_3(x) = 19 - \frac{L}{T} \geq 0$

$g_4(x) = 0.45DWT^{0.31} - T \geq 0$

$g_5(x) = 0.7D + 0.7 - T \geq 0$

$g_6(x) = DWT - 3000 \geq 0$

$g_7(x) = 500000 - DWT \geq 0$

$g_8(x) = 0.32 - F_n \geq 0$

$g_9(x) = KB + BMT - KG - 0.07B \geq 0$

$annual_{costs} = capital_{costs} + running_{costs} + voyage_{costs}$

$capital_{costs} = 0.2ship_{cost}$

$ship_{cost} = 1.3\left(2000W_s^{0.85} + 3500W_o + 2400P^{0.8}\right)$

$W_s = 0.034L^{1.7}B^{0.7}D^{0.4}C^{0.5}$

$W_o = 1.0L^{0.8}B^{0.6}D^{0.3}C_B^{0.1}$

$W_m = 0.17P^{0.9}$

$running_{costs} = 40000DWT^{0.3}$

$DWT = displacement - light_{ship}$

$displacement = 1.025LBTC_B$

$P = \frac{displacement^{\frac{2}{3}} V_k^3}{a+bF_n}$

$F_n = \frac{V}{(gL)^{0.5})}$

$V = 0.5144V_k; g = 9 : 8065$

$a = 4977.06C_B^2 - 8105.61C_B + 4456.51$

$b = -10847.2C_B^2 + 12817C_B - 696032$

$voyage_{costs} = (fuel_{cost} + port_{cost})\,RTPA$

$fuel_{cost} = 1.05 \times \frac{0.19P \times 24}{1000} + 0.2 \times \frac{5000}{24} V_k \times 100$

$port_{cost} = 6.3DWT^{0.8}$

$RPTA = \frac{350}{sea_{days} + port_{days}}$

$sea_{days} = \frac{5000}{24} V_k$

$port_{days} = 2\left(\frac{Cargo_{DWT}}{handling_{rate}} + 0.5\right)$

$Cargo_{DWT} = DWT - \frac{0.19P \times 24}{1000} + 0.2 \times (sea_{days} + 5) - 2DWT^{0.5}$

$handling_{rate} = 8000; KB = 0.53T; KG = 1 + 0.52D$
$BM_T = \frac{(0.085C_B - 0.002)B^2}{TC_B}$
$f_4(x) \; = \sum_{i=1}^{10} max\{g_i(x), 0\}$
where L, B, D, T, V_k, C_B correspond to six decision variables $(x_1, x_2, x_3, x_4, x_5, andx_6)$, respectively. The range of the six variables is as follows: $L \in [150, 274.32]$, $B \in [20, 32.31]$, $D \in [13, 25]$, $T \in [10, 11.71]$, $V_k \in [14, 18]$, and $C_B \in [0.63, 0.75]$

- **Vehicle vibration model problem**
  $f_1(x) =$ seat acceleration
  $f_2(x) =$ forward tire velocity
  $f_3(x) =$ rear tire velocity
  $f_4(x) =$ relative displacement between sprung mass and forward tire
  $f_5(x) =$ relative displacement between sprung mass and rear tire
- **Location of a pollution monitoring system**
  $f(x_1, x_2) \; = -u_1(x_1, x_2) \; - u_2(x_1, x_2) \; - u_3(x_1, x_2)$
  $u_1(x_1, x_2) \; = 3(1 - x_1)^2 e^{-x_1^2 - (x_2+1)^2}$
  $u_2(x_1, x_2) \; = -10(\frac{x_1}{4 - x_1^3 - x_2^5})e^{-x_1^2 x_2^2}$
  $u_3(x_1, x_2) \; = \frac{1}{3} e^{-(x_1+1)^2 - x_2^2}$
  $f_1(x_1, x_2) \; = f(x_1, x_2)$
  $f_2(x_1, x_2) \; = f(x_1 - 1.2, x_2 - 1.5)$
  $f_3(x_1, x_2) \; = f(x_1 + 0.3, x_2 - 3.0)$
  $f_4(x_1, x_2) \; = f(x_1 - 1.0, x_2 + 0.5)$
  $f_5(x_1, x_2) \; = f(x_1 - 0.5, x_2 - 1.7)$
  where $x_1 \in [-4.9, 3.2]$, $x_2 \in [-3.5, 6]$
- **Machining Problem**
  $f_1(x) \; = -7.49 + 0.44x_1 - 1.16x_2 + 0.63x_3$
  $f_2(x) \; = -4.31 + 0.92x_1 - 0.16x_2 + 0.43x_3$
  $f_3(x) \; = 21.90 - 1.94x_1 - 0.30x_2 - 1.04x_3$
  $f_4(x) \; = -11.331 + x_1 + x_2 + x_3$
  $g_1(x) \; = -0.44x_1 + 1.16x_2 - 0.61x_3 \leq -3.17$
  $g_2(x) \; = -0.92x_1 + 0.16x_2 - 0.43x_3 \leq -8.04$
  $g_3(x) \; = -1.94x_1 + 0.30x_2 + 1.04x_2 \leq 18.50$
  $f_5(x) \; = \sum_{i=1}^{10} max\{g_i(x), 0\}$
  where $x_1 \in [6.40, 7.09]$, $x_2 \in [0.69, 2.89]$, and $x_3 \in [3.91, 4.61]$

  Water resource planning
  $f_1(x) \; = 106780.37 \times (x_2 + x_3) \; + 61704.67$
  $f_2(x) \; = 3000x_1$
  $f_3(x) \; = \frac{(305700 \times 2289x_2)}{(0.06 * 2289)^{0.65}}$
  $f_4(x) \; = 2250 \times 2289 \times e^{-39.75x_2 + 9.9x_3 + 2.74)}$
  $f_5(x) \; = 25 \times \left(\frac{1.39}{x_1 x_2} + 4940x_3 - 80\right)$
  $g_1(x) \; = \frac{0.00139}{x_1 x_2} + 4.94x_3 - 0.08 \leq 1.0$
  $g_2(x) \; = \frac{0.000306}{x_1 x_2} + 1.082x_3 - 0.0986 \leq 1.0$
  $g_3(x) \; = \frac{12.307}{x_1 x_2} + 49408.24x_3 + 4051.02 \leq 5000$
  $g_4(x) \; = \frac{2.098}{x_1 x_2} + 8046.33x_3 - 696.71 \leq 16000$
  $g_5(x) \; = \frac{2.138}{x_1 x_2} + 7883.39x_3 - 705.04 \leq 10000$
  $g_6(x) \; = \frac{0.417}{x_1 x_2} + 1721.26x_3 - 136.54 \leq 2000$
  $g_7(x) \; = \frac{0.164}{x_1 x_2} + 631.13x_3 - 54.48 \leq 550$
  $f_6(x) \; = \sum_{i=1}^{10} max\{g_i(x), 0\}$
  where $x_1 \in [0.01, 0.45]$, $x_2 \in [0.01, 0.10]$, and $x_3 \in [0.01, 0.10]$
- **Work roll cooling design problem**
  $f_1(x) \; = \delta TS(x)$
  $f_2(x) \; = S11S(x)$
  $f_3(x) \; = \delta T_{9d}(x)$
  $f_4(x) \; = S11_{9d}(x)$
  $f5(x) \; = \delta T_{15d}(x)$
  $f6(x) \; = S11_{15d}(x)$
  where the decision variables, $x_1 \in [5, 15]$, $x_2 \in [950, 1250]$, $x_3 \in [15, 50]$, $x_4 \in [10, 30]$, $x_5 \in [0.14, 1.256]$, $x_6 \in [40, 80]$, $x_7 \in [20, 100]$
- **Car cab design problem**
  $f_1(x) \; = 1.98 + 4.9x_1 + 6.67x_2 + 6.98x_3 + 4.01x_4 + 1.78x_5 + 0.00001x_6 + 2.73x_7$
  $g_1(x) \; = 1 - (1.16 - 0.3717x_2x_4 - 0.00931x_2x_{10} - 0.484x_3x_9 + 0.01343x_6) \; \geq 0$
  $g_2(x) \; = 0.32 - (0.261 - 0.0159x_1x_2 - 0.188x_1x_8 - 0.019x_2x_7 + 0.0144x_3x_5 + 0.8757x_5x_{10} + 0.08045x_6x_9 + 0.00139x_8x_{11} + 0.00001575x_{10}x_{11}) \; \geq 0$
  $g_3(x) \; = 0.32 - (0.261 - 0.0159x_1x_2 - 0.188x_1x_8 - 0.019x_2x_7 + 0.0144x_3x_5 + 0.8757x_5x_{10} + 0.08045x_6x_9 + 0.00139x_8x_{11} + 0.00001575x_{10}x_{11}) \; \geq 0$
  $g_4(x) \; = 0.32 - (0.74 - 0.61x_2 - 0.163x_3x_8 + 0.001232x_3x_{10} - 0.166x_7x_9 + 0.227x_2x_2) \; \geq 0$
  $g_5(x) \; = 32 - (\frac{URD * MRD * LRD}{3}) \; \geq 0$
  $URD \; = 28.98 + 3.818x_3 - 4.2x_1x_2 + 0.0207x_5x_{10} + 6.63x_6x_9 - 7.77x_7x_8 + 0.32x_9x_{10}$
  $MRD \; = 33.86 + 2.95x_3 + 0.1792x_{10} - 5.057x_1x_2 - 11x_2x_8 - 0.0215x_5x_{10} - 9.98x_7x_8 + 22x_8x_9$
  $LRD = 46.36 - 9.9x_2 - 12.9x_1x_8 + 0.1107x_3x_{10}$
  $g_6(x) \; = 32 - (4.72 - 0.5x - 4 - 0.19x_2x_3 - 0.0122x_4x_{10} + 0.009325x_6x_{10} + 0.000191x_{11}x_{11}) \; \geq 0$
  $g_7(x) \; = 4 - (10.58 - 0.674x_1x_2 - 1.95x_2x_8 + 0.02054x_3x_{10} - 0.0198x_4x_{10} + 0.028x_6x_{10}) \; \geq 0$
  $g_8(x) \; = 9.9 - (16.45 - 0.489x_3x_7 - 0.84x_5x_6 + 0.043x_9x_{10} - 0.0556x_9x_{11} - 0.000786x_{11}x_{11}) \; \geq 0$
  $f_2(x) = max g_1(x), 0$
  $f_3(x) = max g_2(x), 0$
  $f_4(x) = max g_3(x), 0$
  $f_5(x) = max g_4(x), 0$
  $f_6(x) = max g_5(x), 0$
  $f_7(x) = max g_6(x), 0$
  $f_8(x) = max g_7(x), 0$
  $f_9(x) = max g_8(x), 0$
  where $x_1 \in [0.5, 1.5]$, $x_2 \in [0.45, 1.35]$, $x_3 \in [0.5, 1.5]$, $x_4 \in [0.5, 1.5]$, $x_5 \in [0.875, 2.625]$, $x_6 \in [0.4, 1.2]$, $x_7 \in [0.4, 1.2]$
- **Radar waveform design**
  $f_1(x) =$ Median range decodability
  $f_2(x) =$ Median velocity decodability
  $f_3(x) =$ Median range blindness
  $f_4(x) =$ Median velocity blindness
  $f_5(x) =$ Minimum range decodability
  $f_6(x) =$ Minimum velocity decodability

$f_7(x) =$ Minimum range blindness
$f_8(x) =$ Minimum velocity blindness
$f_9(x) =$ Dwell time

- **General aviation aircraft**

$f_1(x) =$ takeoff noise
$f_2(x) =$ empty weight
$f_3(x) =$ direct operating cost
$f_4(x) =$ ride roughness
$f_5(x) =$ fuel weight
$f_6(x) =$ purchase price
$f_7(x) =$ product family dissimilarity
$f_8(x) =$ flight range
$f_9(x) =$ lift/ drag ratio
$f_{10}(x) =$ cruise speed

REFERENCES

[1] Z. Liang, T. Luo, K. Hu, X. Ma, and Z. Zhu, "An indicator-based many-objective evolutionary algorithm with boundary protection," *IEEE Trans. Cybern.*, vol. 51, no. 9, pp. 4553–4566, Sep. 2021.

[2] Z. Zhu, G. Zhang, M. Li, and X. Liu, "Evolutionary multi-objective workflow scheduling in cloud," *IEEE Trans. Parallel Distrib. Syst.*, vol. 27, no. 5, pp. 1344–1357, May 2016.

[3] X. Liu, Y. Zhou, J. Zhao, R. Yao, B. Liu, D. Ma, and Y. Zheng, "Multi-objective ResNet pruning by means of EMOAs for remote sensing scene classification," *Neurocomputing*, vol. 381, pp. 298–305, Mar. 2020.

[4] S. Datta and S. Das, "Multiobjective support vector machines: Handling class imbalance with Pareto optimality," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 5, pp. 1602–1608, May 2019.

[5] S. N. Qasem, S. M. Shamsuddin, S. Z. M. Hashim, M. Darus, and E. Al-Shammari, "Memetic multiobjective particle swarm optimization-based radial basis function network for classification problems," *Inf. Sci.*, vol. 239, pp. 165–190, Aug. 2013.

[6] B. Li, J. Li, K. Tang, and X. Yao, "Many-objective evolutionary algorithms: A survey," *ACM Comput. Surv.*, vol. 48, no. 1, pp. 1–35, 2015.

[7] P. Zhang, J. Li, T. Li, and H. Chen, "A new many-objective evolutionary algorithm based on determinantal point processes," *IEEE Trans. Evol. Comput.*, vol. 25, no. 2, pp. 334–345, Apr. 2021.

[8] Z. Gong, H. Chen, B. Yuan, and X. Yao, "Multiobjective learning in the model space for time series classification," *IEEE Trans. Cybern.*, vol. 49, no. 3, pp. 918–932, Mar. 2019.

[9] P. J. Fleming, R. C. Purshouse, and R. J. Lygoe, "Many-objective optimization: An engineering design perspective," in *Proc. Int. Conf. Evol. Multi-Criterion Optim.* Berlin, Germany: Springer, 2005, pp. 14–32.

[10] H. Chen and X. Yao, "Multiobjective neural network ensembles based on regularized negative correlation learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 12, pp. 1738–1751, Dec. 2010.

[11] H. Chen and X. Yao, "Evolutionary random neural ensembles based on negative correlation learning," in *Proc. IEEE Congr. Evol. Comput.*, Sep. 2007, pp. 1468–1474.

[12] M. Elarbi, S. Bechikh, A. Gupta, L. B. Said, and Y. Ong, "A new decomposition-based NSGA-II for many-objective optimization," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 48, no. 7, pp. 1191–1210, Jul. 2018.

[13] V. Palakonda and R. Mallipeddi, "An evolutionary algorithm for multi and many-objective optimization with adaptive mating and environmental selection," *IEEE Access*, vol. 8, pp. 82781–82796, 2020.

[14] Q. Lin, S. Liu, Q. Zhu, C. Tang, R. Song, J. Chen, C. A. C. Coello, K. Wong, and J. Zhang, "Particle swarm optimization with a balanceable fitness estimation for many-objective optimization problems," *IEEE Trans. Evol. Comput.*, vol. 22, no. 1, pp. 32–46, Feb. 2018.

[15] E. M. N. Figueiredo, T. B. Ludermir, and C. J. A. Bastos-Filho, "Many objective particle swarm optimization," *Inf. Sci.*, vol. 374, pp. 115–134, Dec. 2016.

[16] X.-S. Yang, "Multiobjective firefly algorithm for continuous optimization," *Eng. Comput.*, vol. 29, no. 2, pp. 175–184, Apr. 2013.

[17] T. P. França, L. G. A. Martins, and G. M. B. Oliveira, "MACO/NDS: Many-objective ant colony optimization based on non-dominated sets," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2018, pp. 1–8.

[18] S. Jiang, H. Li, J. Guo, M. Zhong, S. Yang, M. Kaiser, and N. Krasnogor, "AREA: An adaptive reference-set based evolutionary algorithm for multiobjective optimisation," *Inf. Sci.*, vol. 515, pp. 365–387, Apr. 2020.

[19] F. Ming, W. Gong, and L. Wang, "A two-stage evolutionary algorithm with balanced convergence and diversity for many-objective optimization," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 52, no. 10, pp. 6222–6234, Oct. 2022.

[20] 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.

[21] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm," ETH Zürich, Zürich, Switzerland, TIK-Rep. 103, 2001, vol. 103.

[22] E. Zitzler and S. Künzli, "Indicator-based selection in multiobjective search," in *Proc. Int. Conf. Parallel Problem Solving Nature*. Berlin, Germany: Springer, 2004, pp. 832–842.

[23] 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.

[24] T. Pamulapati, R. Mallipeddi, and P. N. Suganthan, "$I_{SDE}+$—An indicator for multi and many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 23, no. 2, pp. 346–352, Apr. 2019.

[25] Y. Sun, G. G. Yen, and Z. Yi, "IGD indicator-based evolutionary algorithm for many-objective optimization problems," *IEEE Trans. Evol. Comput.*, vol. 23, no. 2, pp. 173–187, Apr. 2019.

[26] 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.

[27] R. Wang, R. C. Purshouse, and P. J. Fleming, "Preference-inspired coevolutionary algorithms for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 17, no. 4, pp. 474–494, Aug. 2013.

[28] R. Wang, R. C. Purshouse, and P. J. Fleming, "Preference-inspired co-evolutionary algorithm using weights for many-objective optimization," in *Proc. 15th Annu. Conf. Companion Genet. Evol. Comput.*, Jul. 2013, pp. 101–102.

[29] 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.

[30] Y. Tian, R. Cheng, X. Zhang, Y. Su, and Y. Jin, "A strengthened dominance relation considering convergence and diversity for evolutionary many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 23, no. 2, pp. 331–345, Apr. 2019.

[31] Y. Yuan, H. Xu, B. Wang, and X. Yao, "A new dominance relation-based evolutionary algorithm for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 20, no. 1, pp. 16–37, Feb. 2016.

[32] S. Jiang and S. Yang, "A strength Pareto evolutionary algorithm based on reference direction for multiobjective and many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 21, no. 3, pp. 329–346, Jun. 2017.

[33] L. R. C. de Farias, P. H. M. Braga, H. F. Bassani, and A. F. R. Araújo, "MOEA/D with uniformly randomly adaptive weights," in *Proc. Genet. Evol. Comput. Conf.*, Jul. 2018, pp. 641–648.

[34] J. A. Duro and D. K. Saxena, "Timing the decision support for real-world many-objective optimization problems," in *Proc. Int. Conf. Evol. Multi-Criterion Optim.* Berlin, Germany: Springer, 2017, pp. 191–205.

[35] T. Tušar, "On using real-world problems for benchmarking multiobjective optimization algorithms," in *Proc. Int. Conf. High-Perform. Opt. Ind. (HPOI)*, vol. 500, 2018, pp. 1–4.

[36] O. Cuate and O. Schütze, "Pareto explorer for solving real world applications," *Res. Comput. Sci.*, vol. 149, no. 3, pp. 29–36, 2020.

[37] K. Li, R. Wang, T. Zhang, and H. Ishibuchi, "Evolutionary many-objective optimization: A comparative study of the state-of-the-art," *IEEE Access*, vol. 6, pp. 26194–26214, 2018.

[38] Z.-Z. Liu, Y. Wang, and P.-Q. Huang, "AnD: A many-objective evolutionary algorithm with angle-based selection and shift-based density estimation," *Inf. Sci.*, vol. 509, pp. 400–419, Jan. 2020.

[39] Y. Yuan, H. Xu, B. Wang, B. Zhang, and X. Yao, "Balancing convergence and diversity in decomposition-based many-objective optimizers," *IEEE Trans. Evol. Comput.*, vol. 20, no. 2, pp. 180–198, Apr. 2016.

[40] Y. Yuan, H. Xu, and B. Wang, "Evolutionary many-objective optimization using ensemble fitness ranking," in *Proc. Annu. Conf. Genet. Evol. Comput.*, Jul. 2014, pp. 669–676.

[41] V. Palakonda, J.-M. Kang, and H. Jung, "An adaptive neighborhood based evolutionary algorithm with pivot- solution based selection for multi- and many-objective optimization," *Inf. Sci.*, vol. 607, pp. 126–152, Aug. 2022.

[42] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff, "A reference vector guided evolutionary algorithm for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 20, no. 5, pp. 773–791, Oct. 2016.

[43] H. Jain and K. Deb, "An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach. Part II: Handling constraints and extending to an adaptive approach," *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 602–622, Aug. 2014.

[44] R. Tanabe and H. Ishibuchi, "An easy-to-use real-world multi-objective optimization problem suite," *Appl. Soft Comput.*, vol. 89, Apr. 2020, Art. no. 106078.

[45] M. G. Parsons and R. L. Scott, "Formulation of multicriterion design optimization problems for solution with scalar numerical optimization methods," *J. Ship Res.*, vol. 48, no. 1, pp. 61–76, Mar. 2004.

[46] N. Nariman-Zadeh, M. Salehpour, A. Jamali, and E. Haghgoo, "Pareto optimization of a five-degree of freedom vehicle vibration model using a multi-objective uniform-diversity genetic algorithm (MUGA)," *Eng. Appl. Artif. Intell.*, vol. 23, no. 4, pp. 543–551, Jun. 2010.

[47] A. Jamali, R. Mallipeddi, M. Salehpour, and A. Bagheri, "Multi-objective differential evolution algorithm with fuzzy inference-based adaptive mutation factor for Pareto optimum design of suspension system," *Swarm Evol. Comput.*, vol. 54, May 2020, Art. no. 100666.

[48] K. Miettinen, "Graphical illustration of Pareto optimal solutions," in *Multi-Objective Programming and Goal Programming*. Berlin, Germany: Springer, 2003, pp. 197–202.

[49] L. Thiele, K. Miettinen, P. J. Korhonen, and J. Molina, "A preference-based evolutionary algorithm for multi-objective optimization," *Evol. Comput.*, vol. 17, no. 3, pp. 411–436, Sep. 2009.

[50] M. Ghiassi, R. E. DeVor, M. I. Dessouky, and B. A. Kijowski, "An application of multiple criteria decision making principles for planning machining operations," *IIE Trans.*, vol. 16, no. 2, pp. 106–114, Jun. 1984.

[51] K. Musselman and J. Talavage, "A tradeoff cut approach to multiple objective optimization," *Oper. Res.*, vol. 28, no. 6, pp. 1424–1435, Dec. 1980.

[52] Y. T. Azene, "Work roll system optimisation using thermal analysis and genetic algorithm," Ph.D. dissertation, Cranfield Univ., Cranfield, U.K., 2011.

[53] E. J. Hughes, "Radar waveform optimisation as a many-objective application benchmark," in *Proc. 4th Int. Conf. Evol. Multi-Criterion Optim. (EMO)*. Berlin, Germany: Springer, 2007, pp. 700–714.

[54] T. Simpson, J. Allen, W. Chen, and F. Mistree, "Conceptual design of a family of products through the use of the robust concept extrapolation method," in *Proc. AIAA/USAF/NASA/ISSMO Symp. Multidisciplinary Anal. Optim.*, Sep. 1996, p. 4161.

[55] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach," *IEEE Trans. Evol. Comput.*, vol. 3, no. 4, pp. 257–271, Nov. 1999.

[56] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. D. Fonseca, "Performance assessment of multiobjective optimizers: An analysis and review," *IEEE Trans. Evol. Comput.*, vol. 7, no. 2, pp. 117–132, Apr. 2003.

[57] D. A. Van Veldhuizen and G. B. Lamont, "Evolutionary computation and convergence to a Pareto front," in *Proc. Late Breaking Papers Genet. Program. Conf.* Princeton, NJ, USA: Citeseer, 1998, pp. 221–228.

[58] H. Wang, Y. Jin, and X. Yao, "Diversity assessment in many-objective optimization," *IEEE Trans. Cybern.*, vol. 47, no. 6, pp. 1510–1522, Jun. 2017.

[59] K. Deb and R. B. Agrawal, "Simulated binary crossover for continuous search space," *Complex Syst.*, vol. 9, no. 2, p. 115–148, Apr. 1995.

[60] K. Deb and M. Goyal, "A combined genetic adaptive search (GeneAS) for engineering design," *Comput. Sci. Inf.*, vol. 26, pp. 30–45, Aug. 1996.

[61] J. Alcalá-Fdez, L. Sánchez, S. García, M. J. D. Jesus, S. Ventura, J. M. Garrell, J. Otero, C. Romero, J. Bacardit, V. M. Rivas, J. C. Fernández, and F. Herrera, "KEEL: A software tool to assess evolutionary algorithms for data mining problems," *Soft Comput.*, vol. 13, no. 3, pp. 307–318, Feb. 2009.

[62] U. K. Wickramasinghe, R. Carrese, and X. Li, "Designing airfoils using a reference point based evolutionary many-objective particle swarm optimization algorithm," in *Proc. IEEE Congr. Evol. Comput.*, Jul. 2010, pp. 1–8.

[63] M. Drela and M. B. Giles, "Viscous-inviscid analysis of transonic and low Reynolds number airfoils," *AIAA J.*, vol. 25, no. 10, pp. 1347–1355, Oct. 1987.

VIKAS PALAKONDA received the B.Tech. degree in electronics and communication engineering from the GMR Institute of Technology, affiliated to Jawaharlal Nehru Technological University, Kakinada, Andhra Pradesh, India, in 2015, and the master's and Ph.D. degrees from the School of Electronics Engineering, Kyungpook National University, Daegu, South Korea, in 2017 and 2021, respectively. He was a Postdoctoral Fellow with the Department of Artificial Intelligence, Kyungpook National University. His current research interests include computational intelligence, reinforcement learning, computer vision, complex networks, and multi-objective optimization applications.

JAE-MO KANG (Member, IEEE) received the Ph.D. degree in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2017. He was a Postdoctoral Fellow with the Department of Electrical and Computer Engineering, Queen's University, Kingston, Canada, and an Assistant Professor with the School of Intelligent Mechatronics Engineering, Sejong University, Seoul, South Korea. He is currently an Assistant Professor with the Department of Artificial Intelligence, Kyungpook National University, Daegu, South Korea. His current research interests include the IoT, LoRa, 6G, machine/deep learning, reinforcement learning, federated Learning, and edge AI.