Scientia Iranica D (2021) 28(6), 3293–3314

SCIENTIA IRANICA

Sharif University of Technology
Scientia Iranica
Transactions D: Computer Science & Engineering and Electrical Engineering
http://scientiairanica.sharif.edu

A non-dominated sorting based evolutionary algorithm for many-objective optimization problems

S.U. Mane* **and M.R. Narasinga Rao**

Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation (Deemed to be University), Vaddeswaram, Guntur Dist., AP, India.

Received 25 February 2019; received in revised form 6 December 2020; accepted 17 May 2021

KEYWORDS
Many-objective hybrid differential evolution algorithm;
Non-dominated sorting;
Decomposition-based approach;
Differential evolution algorithm;
Particle swarm optimization algorithm;
Many-objective optimization problems.

Abstract. The optimization problems with more than three objectives are Many-objective Optimization Problems (MaOPs) that exist in various scientific and engineering domains. The existing multi-objective evolutionary algorithms are not found effective in addressing the MaOPs. Its limitations initiated the need to develop an algorithm that efficiently solves MaOPs. The proposed work presents the design of the Many-Objective Hybrid Differential Evolution (MaOHDE) algorithm to address MaOPs. Initially, two multi-objective evolutionary algorithms viz. Non-dominated Sorting based Multi-Objective Differential Evolution (NS-MODE) and Non-dominated Sorting based Multi-Objective Particle Swarm Optimization (NS-MOPSO) algorithms were designed. These algorithms were developed by incorporating the non-dominated sorting approach from Non-dominated Sorting-based Genetic Algorithm II (NSGA-II), the ranking approach, weight vector, and reference points. Tchebycheff–a decomposition-based approach, was applied to decompose the MaOPs. The MaOHDE algorithm was developed by hybridizing the NS-MODE with the NS-MOPSO algorithm. The strength of the presented approach was determined using 20 instances of DTLZ functions, and its effectiveness and efficiency were verified upon its comparison with the recently developed state of algorithms existing in the literature. From the results, it is observed that the MaOHDE responds better than its competitors or is competitive for most of the test instances and the convergence rate is also good.

1. Introduction

Optimization problems are naturally found in various engineering, scientific, and business domains, each enjoying different nature. In the real world, it is common to administer optimization problems with three or more objectives; such problems are associated with "Many-objective Optimization Problems (MaOPs)" group and obtaining an optimal solution to such problems remains a challenging task. The main components required to design effective many-objective optimization algorithms include the aggravation of conflicting objectives, convergence, and diversity along with the requirement of objective space [1]. Several approaches have been explored in the literature that address the MaOPs [2]. In Multi-objective Optimization Problems (MOPs), two or more clashing objective functions are optimized simultaneously. These clashing objective functions limit the optimization algorithms to obtaining a unique optimal solution simultaneously for all of them. The "Pareto-optimum" group of obtained findings shows the balance among diverse objective functions, also known as "Pareto-optimal solutions". It is known as "Pareto Front (PF)" and

*. *Corresponding author.*
E-mail addresses: manesandip82@gmail.com (S.U. Mane); ramanarasingarao@kluniversity.in (M.R. Narasinga Rao)

doi: 10.24200/sci.2021.53026.3017

"Pareto Set (PS)" in the search and decision spaces, respectively [3]. Researchers are working on the development of Multi-Objective Evolutionary Algorithms (MOEAs). MOEAs are successfully applied to address diverse MOPs, but MOEAs face many problems while solving MOPs beyond three objective functions [2].

The challenges faced by MOEAs to solve MaOPs are summarized in [1–7]. These challenges increased pressure on the development and use of evolutionary algorithms for MaOPs along with the application of existing MOEAs. Also, the MOEAs are not scalable to address the MaOPs [8,9]. Challenges to solve MaOPs are as follows:

1. The obtained results become non-dominated when the number of objective functions increases;
2. Confliction between diversity and convergence is aggravated when objective space in size increases;
3. For computational efficiency, population size can be small;
4. Computational complexity grows exponentially, while the number of objectives increases (e.g., Hypervolume (HV) calculation);
5. Balancing diversity and convergence is much more difficult;
6. Visualization of the Pareto-optimal front is difficult due to large dimensions. This issue may not have a direct effect on the evolutionary process, but it causes difficulty for the designer/ authority to select the ideal result.

There are mainly three approaches to solving MaOPs, viz., convergence enhancement approach, decomposition-based approach, and performance indicator-based approach [10]. The "Pareto, α, ε, and cone ε" dominance criteria are found in the literature, as well [10]. These are evaluated on a family of scalable, convex multi-objective minimization problems. Other dominance criteria found in the literature are "fuzzy dominance, contraction or expansion of dominance area, Pareto-adaptive ε- dominance, volume dominance, and L-dominance". Among these criteria, "Pareto dominance" is largely used. As the number of objective functions increases, the ordering of "Pareto dominance" degrades [10].

Trivedi et al. [11] conducted a comprehensive survey of decomposition-based evolutionary approaches designed to solve bi- or tri-objective problems. They categorized MOEAs into dominance-based, indicator-based, and decomposition-based approaches. The dominance-based approaches are inappropriate for MaOPs due to the large size of objective functions. The indicator-based approaches use HV indicator, whose computational cost raises largely due to the large size of objective functions. Decomposition-based approaches have become popular since 2007. The design components of decomposition-based evolutionary algorithms are "(1) weight vector generation method, (2) decomposition method, (3) computational resource allocation strategy, (4) mating selection mechanism, (5) reproduction operators, and (6) replacement procedure" [11].

The major contribution of the proposed work is to design a hybrid many-objective optimization algorithm. This study presents the design of two multi-objective evolutionary approaches, namely Non-dominated Sorting based Multi-Objective Differential Evolution (NS-MODE) and Non-dominated Sorting based Multi-Objective Particle Swarm Optimization (NS-MOPSO) algorithms. Both of the proposed approaches incorporate the non-dominated sorting scheme from NSGA-II, the weight vector to assign a weight to each objective, reference points to compare and select the solutions, and a ranking approach in order to rank the non-dominated solutions. The decomposition-based approach is used to simplify MaOPs. The Many-Objective Hybrid Differential Evolution (MaOHDE) algorithm is developed by combining the NS-MODE and NS-MOPSO algorithms. The applicability of the proposed approaches is verified using well-known DTLZ test functions on different objectives.

The main objectives of the presented work are:

- To design NS-MODE and NS-MOPSO algorithms;
- To effectively use non-dominated sorting, weight vector, reference points, and ranking scheme to develop the proposed approach;
- To design the Many-Objective Evolutionary Algorithm (MaOEA) as a MaOHDE algorithm by combining NS-MODE and NS-MOPSO algorithms;
- To evaluate the performance of the proposed algorithms using the Inverted Generational Distance (IGD), HV and average performance score metrics in dealing with twenty instances of DTLZ test functions.

The remaining paper is organized as follows. Section 2 presents a literature review of MaOPs and problem-solving strategies. The standard or traditional Differential Evolution (DE) and Particle Swarm Optimization (PSO) algorithms, as well as hybridization of the DE algorithm with its applications, are also introduced in this section. Section 3 presents the proposed methodology to address MaOPs. Section 4 gives the obtained results and discussion. Section 5 presents the conclusions of the study and further research directions.

2. Literature review

This section outlines the findings existing in the literature about MaOPs and different problem-solving strategies. The many-objective or multi-objective problem-solving strategies are particularly focused in this section. This section also introduces the basic DE algorithm, PSO algorithm, and hybrid DE algorithm.

The optimization problems in various scientific and engineering domains are modelled based on various characteristics and nature of the problems. The optimization problems can be continuous or discrete based on the values of optimization functions. If the value of the objective function depends on certain conditions, such problems are known as constrained optimization problems, otherwise unconstrained optimization problems. Also, a number of objective functions are used to categorize the optimization problems. If the number of objective functions is one, two, and three or more than three, the problems are known as single-objective, bi-objective, and MOPs or MaOPs, respectively. Researchers have introduced various classical and nature-inspired or non-traditional techniques to solve single-, bi-objective and MOPs [12–16].

2.1. Many-objective Optimization Problems (MaOPs)

The MaOPs are difficult to solve and an optimal solution is obtained using Evolutionary Multi-Objective Algorithms (EMOAs). The challenges faced by EMOAs when dealing with a large set of objective functions were discussed in [1,3,6,7]. Giagkiozis and Fleming [4,5] investigated the decomposition method. The 'Chebyshev scalarizing function' is used over 'Pareto-based' methods. The scaling of a large-scale problem and the convergence rate of the decomposition method are better than those of the Pareto-based methods [4,5]. MaOPs are mostly solved using decomposition methods [4–6]. The scalarizing functions are used by decomposition-based methods to generate a set of single-objective problems from many-objective problems. Evolutionary algorithms with the Pareto-based methods face difficulties like poor convergence rate due to increase in the number of dimensions and poor diversity in the solution. The Chebyshev scalarizing function and Pareto dominance methods are equivalent [4–6]. Li et al. proposed the MOEA/DD algorithm that combines decomposition and dominance-based approaches [3]. The MOEA/DD uses weight vectors to identify the numerous sub-regions in the feasible region. After experimentation, the superiority of MOEA/DD was observed in dealing with such problems. The EAs are designed to maximize the diversity and minimize the distance between function evaluations and solutions (i.e., convergence) in MaOPs. The key issue is to attain the balance between diversity and convergence in the case of such EAs types [3,6,7].

The reference vectors are utilized to decompose the MaOPs in multiple single-objective sub-problems as well as to make it clear that users prefer the subset of PF [2]. Angle penalized distance, which is a scalarization approach, is used to make a balance between the diversity and convergence of solutions. The Reference vector guided Evolutionary Algorithm (REVA) adopts the elitism strategy similar to the NSGA-III algorithm [17]. Reference vector adoption and reference vector-guided selection are used in REVA for elitism strategy. The four-step reference vector-guided selection scheme is used to select the solution in each subspace separately, where subspace is formed by partitioning the objective space into multiple subspaces [2]. The suitability of the REVA is evaluated using six DTLZ and nine WFG functions. With these modifications and after comparison with 'MOEA/DD, NSGA-III, MOEA/D-PBI, GrEA, and KnEA', REVA exhibits a robust performance [2]. Dai et al. [18] used the M2M population decomposition strategy to improve the k-dominated sorting that enhances the population diversity and convergence and avoids circular dominance. M2M population decomposition strategy uses a set of the uniformly distributed units to split the population into sub-problems. The solutions in the subpopulations are ranked to select new populations in the next generation using improved k-dominance. The relationship of dominance between individuals is evaluated in terms of best and worst objectives. For this reason, an individual's relationship with dominance is no longer affected by the objective dimension. The k-dominance cannot take the objective value into account. The four DTLZ functions are used to compare the results with the NSGA-III algorithm [18].

Mohammadi et al. [1] integrated the user preferences with decomposition strategies to propose a new R-MEAD2 algorithm, which improves the scalability of the R-MEAD algorithm. The dimension of the objective determines the size of the population and it is due to the weight vectors created using a simplex-lattice design method. The R-MEAD2 removes this dependency using a 'uniform random number generator'. The 'Tchebycheff and PBI' are used to compare the performances of R-MEAD2 and R-NSGA-III for DTLZ functions [1].

Pal et al. [19] presented 'a correlation distance-based automatic objective reduction algorithm', based on which the PS is clustered. The objective values, other than those present at the centre from the dense cluster, are removed. The searching operation is performed on a reduced set. The algorithm's execution process ends as soon as the total cluster and objectives become equal. Therefore, the performance is improved in terms of faster convergence, exploration, and exploitation. The clustering approach is based on the k-medoid clustering algorithm. The DEMO algorithm is used to solve the MaOPs with a reduced objective set. The DTLZ1-DTLZ4 functions used to evaluate the algorithm for 10 and 20 objectives [19]. This framework has few limitations viz. it works when the relation among objective pairs is linear, the singleton clusters are not detected, and it fails to maintain true variety or even distribution of the solutions. The elimination of such limitations could be the topic of future research work [19].

Marler and Arora [9] presented a survey of 'multi-objective optimization methods' developed to address multi-objective engineering problems. After this study, it is found that there is no single method that is superior to obtain optimum results for MOPs from a single domain [9]. Ishibuchi et al. [20] presented a diminutive review of the application of evolutionary algorithms to address MaOPs. The authors discussed difficulties while solving MaOPs using EMOAs. The scalability improvement approaches explained are used to improve the performance of existing EMOAs. Authors suggest the hybridization of many-objective optimization algorithms; therefore, there will be a focused search for part of the PF and global search in the entire PF. The representation of the obtained solution is also a challenging task in MaOPs. The authors have also listed many objective test problems. One of the future research scopes presented is about the use of alternative performance indicators to HV as well as a reduction in the computation cost of HV [20]. Lücken et al. [21] presented a survey of the application of MOEAs to address MaOPs, use of performance measurement indicators and challenges while solving MaOPs and future research directions. Authors have surveyed different approaches based on MOEAs. There are mainly two methods, preference relation based and original problem transformation [21]. The performance metrics preferred by researchers are HV, coverage, Generational Distance (GD), IGD, convergence, etc. The HV is widely used; however, its computational cost is high. The algorithm's performance generalization is hard because it gets changed as the number of objectives increases or decreases, changes in algorithms and common parameter settings, etc. Most of the researchers have used the DTLZ test suit. The problems faced by a single method to solve MaOPs can be overcome by hybridizing the two different methods [21]. A short review of MaOPs, algorithms to solve MaOPs, and research challenges was given by Mane and Narasinga Rao [12].

The recent development of MaOEAs has focused on the improvement of various limitations of existing MoEAs. The increase in the number of objectives increases the complexity of the problem. Sometimes, the MaOPs contain redundant objectives and authors work to improve the performance of MaOEAs by eliminating less important redundant objectives [22]. Objective extraction is another strategy used to convert the MaOPs into MOPs. The fuzzy clustering strategy is used to develop such an approach [23]. The MaOEAs behave differently when applied to solve MaOPs with a different set of features. It is a challenging task to adjust the algorithms working strategy to work well for problems with different features. The learning automation is incorporated with decomposition-based MaOEAs [24]. The adaptive weighted decomposition strategy is used to develop the role of MaOEAs in addressing the problems with regular and irregular Pareto-fronts [25]. The multi-stage MaOEA proposed to make a balance between convergence and diversity while solving MaOPs. The first stage considers the convergence of solutions, while the second stage maintains the diversity among a set of solutions [26]. To use the advantages of indicator-based and decomposition-based strategies, authors have developed R2-indicator and decomposition-based evolutionary algorithms [27]. The adaptive clustering mechanism is used to improve the selection of the most preferable solution and opposition-based learning used to initialize the solution while designing MaOEAs to solve the MaOPs. The selected strategies balance both convergence and diversity [28]. The diversity improvement without affecting the convergence of MaOEAs is implemented by performing the region-based search [29]. Li et al. presented the comparison of thirteen many-objective algorithms categorized into decomposition-based, indicator-based, diversity-based selection modification, dominance relation modification, and preference-based approaches [30]. Authors selected web-of-science-indexed publications from 2003 to 2016 to select the algorithms. The main purpose of the comparison is to guide researchers in choosing an appropriate algorithm to address the selected problem. The recent development of MaOEAs was presented by Mane and Narasingrao [31]. The chaotic-based improved many-objective Jaya algorithm was developed to address the MaOPs [31].

The researchers have discussed various factors in developing MaOPs-solving approaches. Researchers have identified the difficulties and limitations of existing multi-objective optimization algorithms while solving MaOPs. The literature study of open-up important factors is focused while developing MaOP-solving approaches. The new researchers can use the hybrid approach, where two global or global-local approaches can be combined. The performance metric can be used to develop a mathematical model of selected MaOPs.

2.2. Problem-solving strategies used to solve MaOPs

Several approaches have proposed augmenting the functioning of MOEAs. The most widely used approaches found in the literature are:

1. **Convergence enhancement approach.** The convergence rate is enhanced by performing selection towards PF or combining the convergence related performance metric with the Pareto-dominance. Such modifications are found in [2];
2. **Performance indicator-based approach.** This type of approach is developed using various performance metrics, limiting the drawback of dominance-based approaches [2];
3. **Decomposition-based approach.** This approach is used to transform the complex MaOPs into a set of 'single-objective' optimization problems so that they can be solved collaboratively [1,2]. The 'weighted Tchebycheff approach, the weighted sum approach, and the PBI approach are widely used [1,6]. The applications of such approaches are found in [2–5,18].

   The Tchebycheff, a decomposition-based approach, is mathematically presented in Eq. (1) taken from [1]:

$$\text{minimization} = (g^{tch}(x, w, z^*) \\ = \max\{W_i|f_i(x) - Z_i^*|\}), \qquad (1)$$

   where $Z^*\hat{I}R^m = ideal\ points$, $w = (w_1, \cdots, w_m)$ is positive weight vector. There is one weight-vector 'w' present for each Pareto-optimal point 'x^*' such that it is an optimal solution to Eq. (1).

   Other than the approaches discussed here, few other approaches can be used to enhance the performance of MOEAs in solving MaOPs. These are reference points, interactive user preferences, reduced number of objectives, etc. [32,33].
4. **Reference vector.** The reference points help decision-makers search for the preferred set of solutions. It guides the selection process of the algorithm to find a single or a set of solutions located near the reference point. The reference points help emphasize the solutions that are close to the reference points [18,34]. The two to five reference points are selected by researchers to resolve two to ten objective problems [17,34]. The reference points are used to escort the examining process towards the Pareto-front as well as to explore the search space effectively. Some of the researchers have employed a fixed set of reference points and structured on a normalized hyperplane by a systematic approach [33]. The intention behind the selection of this problem-solving strategy found in [1,34] is as follows:
   1. The feasible results that are closer to the clues are greatly stressed;
   2. It helps the decision-maker to focus on the smaller (i.e., preferred) region of the PF;
   3. It performs the guided and focused search instead of approximating the entire PF.

The literature study helps determine the research gap or future research directions. The MaOPs can be decomposed using various decomposition techniques. The decomposition reduces the complexity of problems. The weighted Tchebycheff method is widely used to decompose MaOPs. The non-dominated sorting using the ranking approach gives good results when used to solve MOPs. The reference point vector used by researchers helps give search direction towards the Pareto optimal region and improves the search speed. The IGD performance metric measures the convergence to the PF as well as the diversity of solutions. The computational cost of the IGD performance metric is lower than that of other performance metrics used to measure the quality of solutions in MaOPs.

2.3. Standard DE, PSO, and hybridization of the differential evolution algorithm

The standard DE algorithm, PSO algorithm, and hybridization of the DE algorithm are briefly presented here. Applications of DE and PSO algorithms are cited in this section. Different ways of hybridization of the DE algorithm from literature are also listed.

2.3.1. Standard DE algorithm

It is a stochastic evolutionary algorithm developed for addressing optimization problems. The theoretical framework of DE is simple and researchers or programmers need to tune fewer algorithm-specific parameters. The DE algorithm is characterized by the following advantages: the solution is enhanced immediately after each iteration and it is capable to address objective functions with various characteristics, convergence property is good, easy to parallelize, and ease to use [35]. The new population in the DE algorithm is generated in a repeated cycle using the three operators, viz., mutation, crossover, and selection. The key process in the DE algorithm is to generate the trial vector; the trial vector is generated according to the steps given by Zhao et al. [36]. Kachitvichyanukul [37] compared DE, PSO and Genetic Algorithm (GA) with each other, as presented in Table 1.

As compared to GA and PSO, the DE algorithm has two main advantages:

1. Improvement takes place in the solution immediately after each iteration;
2. Requirement of a smaller number of algorithm-specific parameters.

Various applications of the DE algorithm are found in the literature. Das et al. [38] presented a critical

Table 1. Comparison of Differential Evolution (DE), Particlr Swarm Optimization (PSO), and Genetic Algorithm (GA).

Parameter	GA	PSO	DE
Ranking	Yes	No	No
Influence of population size on the solution time	Exponential	Linear	Linear
Best solution influence on population	Median	Most	Less
Average fitness that cannot get worse	False	False	True
Premature coverage tendency	Median	High	Low
Search space continuity	Less	More	More
Without the local search ability to reach a good solution	Less	More	More
Improving the convergence by homogeneous sub-grouping	Yes	Yes	NA

survey of the DE algorithm. The authors presented applications of science and technology, hybridization of the DE algorithm, and future research directions. Other researchers have quoted that since 2011 to 2015, SCI database contains 4750 research articles related to the DE algorithm [38]. More details about the DE algorithm can be found in [35].

2.3.2. Standard PSO algorithm

It is a swarm intelligence-based approach that is motivated by the communal behavior of birds or fish in the hunt for food. The PSO has a stochastic search scheme and it is capable of rapid convergence and simple computation. It maintains a swarm size that represents the candidate solution [39]. Evolutionary algorithms have the selection operator to select individuals for the next iterations, but the PSO algorithm does not have a selection operator; all the swarms used in the optimization process do not die out [39]. The initialization in PSO is done by generating random particles. The iterative search is performed to find the optimum result. The PSO algorithm contains velocity, position, inertia weight, and constriction factor parameters that need to be tuned properly to obtain the best results [39].

Each particle has its corresponding velocity as well as the fitness value. The direction of the particle movement is provided by the velocity [39,40]. In each iteration, the movement of an ith particle depends on two key factors – the first is the global best position (g) and the second is the local best position (V_i) [39]. The particle's velocity is not only an important mechanism for the particle's movement but also to make the balance between exploration and exploitation. To control the trade-off between local and global search, the inertia weight factor plays an important role. When the inertia weight factor is high, the particles explore global search space, whereas the low inertia weight factor encourages particles to search in the local search space [39]. Song and Guo [41] presented a review of research studies about PSO and mentioned various application domains, variations of PSO, and algorithmic improvement. The different variations of the PSO algorithm as well as applications in different domains have been discussed by the researchers [41–47].

2.3.3. Hybridization of DE algorithm

Hybridization is one of the ways to improve the performance of global search techniques. There are different ways to hybridize any global search technique, viz., global-global, global-local, and global-global-local approaches. Dragoi and Dafinescu [48] presented a survey of the hybridization approaches of the DE algorithm. The DE algorithm is hybridized with swarm intelligence techniques, among which PSO is the preferred choice. The DE algorithm is also hybridized with evolutionary algorithms and other local search techniques. Authors conclude that problem-specific factors to be improved decide the hybridization of DE locally or globally. Das et al. [38] presented the hybridization of the DE algorithm with global and local search methods. Some of the applications listed by Das et al. not only hybridized the DE algorithm but also incorporated parameter adaptation techniques [38]. The authors listed a few hybrid approaches of DE developed to address single objective and MOPs. The DE algorithm is hybridized with PSO, ant colony search, multiplier updating, simulated annealing, adaptive Gaussian immune algorithm, flower pollination algorithm, back-propagation algorithm, set-based approach, biogeography-based optimization, Lagrangian relaxation method, greedy approach, and many more algorithms or methods [38]. Also, the researchers have developed a hybrid DE algorithm to address various optimization problems [49–52]. The hybridization of other nature-inspired algorithms used to address various optimization as well as other problems from different domains. The penalty-guided biogeography-based optimization algorithm is presented to solve the reliability optimization problem with nonlinear resource constraints [53]. The hybrid PSO-GA algorithm is developed to solve constrained optimization problems. Exploration and exploitation are balanced by integrating crossover and mutation operators in the PSO algorithm [54]. Patwal et al. presented TVAC-PSO algorithm with mutation strategies for generation scheduling of pumped storage hydrothermal system [55].

The literature study of the DE algorithm reveals some important points. The solution obtained by the DE algorithm is improved after the completion of each iteration. If the initial population is generated by other search techniques and given input to the DE algorithm, better results can be obtained. This process is controlled initialization of the DE algorithm. The initial population can be controlled using one of the search approaches found in the literature. To improve the performance of the DE algorithm, different mutation or crossover operators are used or hybridized with other search techniques. The PSO algorithm is one of the best search techniques that can be used to generate the initial population for the DE algorithm. The best search results obtained by the PSO algorithm can be improved by the DE algorithm in the latter stages.

3. Proposed approach

This section presents the design of NS-MODE and NS-MOPSO algorithms and their use to develop a hybrid many-objective optimization algorithm called MaO-HDE with its major steps. The proposed approaches presented in this paper are inspired by a survey of the literature. The literature identifies the limitations of existing MOEAs to solve MaOPs. Non-dominated sorting, decomposition of complex problems, use of reference vector, and assigning weight to objectives make the MaOP solving quite easier. These components used to develop the proposed approach make the proposed approach stronger. Also, the advantages of standard DE and PSO algorithm motivated the integration of NS-MODE and NS-MOPSO (global-global search technique integration strategy) to develop the proposed hybrid algorithm. The research challenges discussed by various researchers in the literature to solve MaOPs provide the background for the proposed study.

3.1. NS-MODE algorithm

The proposed NS-MODE algorithm is developed by integrating 'non-dominated sorting and ranking from NSGA-II', 'reference vector', and 'weight vector'. The general procedure of the proposed NS-MODE algorithm consists of three major parts:

1. **Initialization:** This part generates an initial solution randomly;
2. **Evaluation:** This part evaluates generated solution in terms of reference points and weight vectors;
3. **Evolution:** Offspring are generated during this part using non-dominated sorting and ranking scheme.

Algorithm 1 presents the steps of the NS-MODE algorithm. The working of the NS-MODE algorithm is described as follows:

Step 1: *Initialization.* Initial population P_0 is generated randomly with N individuals and the weight vectors and reference points are defined;

Step 2: *Main Cycle.* The population is evolved, evaluated, and updated using the non-dominated sorting and ranking method.

The better solution is evolved from the initial population using crossover and mutation operators. The 'evolution' function first decomposes the MaOPs $f(x)$ into multiple single-objective functions. Next, it uses

Input: number of generation $G_{DE\max}$, Number of objectives M, the number of individuals N, reference points R_p, weight vector W.
Output: Final population $P_{\max}$
Algorithm
Initialization: generate an initial population with N individuals
Evaluate $(P_o,\ W,\ R_p)$
While $t < G_{DE\max}$
Q_t = Evolve $(P_t,\ W_t,\ R_p)$
Evaluate $(Q_t, f(x))$
$U_t = P_t \cup Q_t$
Non-dominated Sorting (U_t, R_p)
Ranking (U_t)
Select the first N individuals (P_t) from ranked individuals (U_t) and go to the next iteration.
End while
End

Algorithm 1. Non-dominated Sorting based Multi-Objective Differential Evolution (NS-MODE).

the DE mutation operator to generate a Q_t population, which is assessed using the decomposed objective function, to find a set of solutions.

The solutions P_t (from the previous iteration) and Q_t (from the current iteration) are combined in U_t and the non-dominated sorting is applied to sort the combined solutions U_t. For each reference point, the Euclidian distance is calculated between each solution of a front and solutions are sorted in ascending order. Each solution is ranked according to the sorted solutions. At the end of the iteration, the first N solutions from ranked solutions are selected for the next iteration. The algorithm stops when it attains the predefined maximum number of iterations.

The NS-MODE algorithm is not a completely new algorithmic approach; rather, it incorporates the non-dominated sorting approach and ranking scheme so that the proposed approach effectively addresses the MaOPs. The traditional DE algorithm is developed by researchers to solve single-objective optimization problems. The purpose of modifying the existing DE algorithm is to develop a suitable method for MaOPs.

3.2. NS-MOPSO algorithm

Algorithm 2 presents the steps of the non-dominated sorting based MOPSO algorithm, for MaOPs. The proposed NS-MOPSO algorithm uses the 'non-dominated sorting and ranking method'.

The main steps for NS-MOPSO are outlined below:

Step 1: *Initialization.* Initial population P_0 is generated randomly with N particles which have random velocity and position. Inertia weight is assigned. The weight vectors and reference points are defined;

Step 2: *Main cycle.* The population is evolved, evaluated, and updated using the non-dominated sorting and ranking method. The evolution function first decomposes the MaOPs, $f(x)$ into multiple single-objective functions. It finds the best value for each particle. With the velocity and position of each particle, PSO generates a Q_t population. The newly generated population Q_t is assessed using the decomposed objective function to find a set of solutions. The solutions P_t and Q_t are combined in U_t. The non-dominated sorting function operated on U_t to sort the population. For each reference point, the Euclidian distance is calculated between each solution of the front and sorted in ascending order. The sorted solutions are ranked. The first N solutions from ranked solutions are selected for the next iteration. The process continues until the algorithm reaches the predefined maximum number of iterations.

The PSO and MOPSO algorithms are effective in solving single- and multi-objective optimization problems, respectively. To address the MaOPS, the non-dominated sorting approach with a ranking scheme is incorporated while designing the NS-MOPSO algorithm. This modification helps the MOPSO algorithm deal with three or more objectives.

3.3. MaOHDE algorithm

The steps for MaOHDE algorithm are presented in Algorithm 3. It consists of mainly two phases. Phase I executes the NS-MOPSO algorithm. In Phase II, the solutions obtained by NS-MOPSO are given as input to the NS-MODE algorithm. The NS-MODE algorithm

1) Input: number of generation $G_{PSO\max}$, number of objectives M, number of particles N with the random velocity and position, reference points, weight vector
2) Output: Final population $P_{\max}$
3) Algorithm
4) Initialization: Generate initial population P_0 with N particles which have random velocity and position. Inertia weights, weight vectors (W_i), and reference points (Rp) are initialized.
5) Evaluate (P_o, W, R_p)
6) While $t < G_{PSO\max}$
7) $Q_t = \text{Evolve}(P_t, W_t, R_p)$
8) Evaluate $(Q_t, f(x))$
9) $U_t = P_t \cup Q_t$
10) Non-dominated Sorting (U_t, R_p)
11) Ranking (U_t)
12) Select the first N individuals (P_t) from ranked individuals (U_t) and go to the next iteration.
13) End while
14) End

Algorithm 2. Non-dominated Sorting based Multi-Objective Particle Swarm Optimization (NS-MOPSO).

1) Input: number of generation $G_{PSO\max}, G_{DE\max}$, number of objectives M , number of particles N with the random velocity and position, reference points, weight vector
2) Output: Final population $P_{\max}$
3) Algorithm
4) Initialization: Generate initial population P_0 with N particles which have random velocity and position. Inertia weights, weight W_i vectors, and reference points Rp are initialized
5) Evaluate (P_o, W, R_p)
6) While $t < G_{PSO\max}$
7) Q_t = Evolve $(P_t,\ W_t,\ R_p)$
8) Evaluate $(Q_t, f(x))$
9) $U_t = P_t \cup Q_t$
10) Non-dominated Sorting (U_t, R_p)
11) Ranking (U_t)
12) Select the first N individuals (P_t) from ranked individuals (U_t) and go to the next iteration.
13) End while
14) Initialization: copy the last iteration population of NS-MOPSO as an initial population P_0 of NS-MODE.
15) Evaluate $(P_o,\ W,\ R_p)$
16) While $t < G_{DE\max}$
17) Q_t = Evolve (P_t, W_t, R_p)
18) Evaluate $(Q_t, f(x))$
19) $U_t = P_t \cup Q_t$
20) Non-dominated Sorting (U_t, R_p)
21) Ranking (U_t)
22) Select the first N individuals (P_t) from ranked individuals (U_t) and go to the next iteration.
23) End while
24) End

Algorithm 3. Many-Objective Hybrid Differential Evolution (MaOHDE).

improves the obtained solution during Phase I. The main steps for MaOHDE are outlined below:

Step 1: *Initialization.* Initial population P_0 is generated randomly with N particles, which have random velocity and position. Termination criteria $G_{DE_{\max}}$ and $G_{PSO_{\max}}$ (both are equal) and inertia weights are assigned. The weight vectors and reference points are initialized;

Step 2: *Main cycle.* The evaluation process is similar to the NS-MODE and NS-MOPSO algorithms. The evaluation process of MaOHDE is laid out as NS-MOPSO algorithm is executed until termination criterion $G_{PSO_{\max}}$ is satisfied. The solution obtained at the end of Phase I is used as an initial population of the NS-MODE algorithm. During Phase II, the NS-MODE algorithm improves the solution and terminates when termination criterion $G_{DE_{\max}}$ is satisfied.

The MaOHDE algorithm consists of two phases. The first phase executes the NS-MOPSO algorithm to produce the initial population randomly, which is inputted to the second phase. Many researchers whose works appear in the literature have discussed how to control the initial population of the DE algorithm. To control the initial population, the NS-MOPSO algorithm is utilized in the proposed approach. The second phase consists of an NS-MODE algorithm. The initial population for the NS-MODE algorithm is inputted from the final population of NS-MOPSO algorithm, which is not randomly produced. At the second phase of MaOHDE, the NS-MODE algorithm does not generate the random population. The NS-MODE evaluates and updates the initial population copied from the NS-MOPSO algorithm.

The two global search techniques can be combined to obtain better results. The approach proposed in this paper consists of a combination of two global search techniques. The DE algorithm gives better results when the initial population is generated by some local or global search technique. If the randomly generated initial population of the DE algorithm is controlled, the DE algorithm works effectively and efficiently [48]. The proposed NS-MOPSO algorithm is used to control the initial population for the NS-MODE algorithm. The solution obtained by NS-MOPSO at the first phase which is inputted as an initial population to NS-MODE at the second phase helps perform a guided search to obtain a better solution. The proposed MaOHDE algorithm aims to mend the convergence rate and maintains diversity.

The major improvement of the proposed approach is presented here. The NS-MODE algorithm is an enhanced version of the standard DE algorithm. The standard DE algorithm is one of the powerful algorithms to solve single-objective optimization problems. As the enhanced version is developed by integrating various components from other algorithms, its capability is improved to address MOPs. As compared to other evolutionary algorithms, the NS-MODE algorithm requires fewer algorithm-specific parameters. Another multi-objective algorithm proposed is NS-MOPSO. The standard PSO algorithm has important features including good convergence rate, balanced exploration and exploitation, and diversity among solutions. However, when the problem's dimensionality increases, the performance of standard PSO degrades. The proposed NS-MOPSO algorithm is an enhanced version of the standard PSO algorithm developed by integrating different components from other MOEAs so that it becomes capable to address MOPs. The researchers have found that when the standard DE algorithm is inputted as a controlled initial solution, its performance is improved. The MaOHDE algorithm uses this feature. The NS-MOPSO algorithm is used to generate an initial population for the NS-MODE algorithm. The proposed design scheme differentiates the MaOHDE algorithm from other MaOEAs. The integration of two global search techniques helps improve the quality of the obtained solutions. The MaOHDE algorithm is designed specifically by considering MaOPs. The decomposition strategy reduces the complexity of the MaOPs and obtains a solution collaboratively by solving the number of single-objective optimization problems. It improves the competence of the proposed approach.

The limitation of the proposed MaOHDE approach is that it requires some iterations equal to two times that of the NS-MODE or NS-MOPSO algorithm. This limitation results from the use of an equal number of iterations for the execution of Phase I (NS-MOPSO) and Phase II (NS-MODE) of the MaOHDE algorithm. The diversity among individuals was affected due to the use of the output from Phase I to the NS-MODE algorithm as the input.

3.4. Time complexity analysis

The time complexity analysis of the proposed approach is presented here. From Algorithm 3, the proposed MaOHDE algorithm consists of mainly two phases. The NS-MOPSO performs a search two times to update the solution. The time complexity of the NS-MOPSO algorithm is $O(MN^2)$. The NS-MODE algorithm updates the solution within a single phase. The time complexity of the NS-MODE algorithm is $O(MN)$. The MaOHDE algorithm at Phase I executes the NS-MOPSO and the NS-MODE algorithm is executed at Phase II. The time complexity of MaOHDE is $O(MN^2)$.

4. Experimental results and discussion

This section presents the experimentation performed by the proposed approaches viz. NS-MOPSO, NS-MODE, and MaOHDE algorithm. First, the benchmark test problems, performance indicator, and parameter settings are presented. Next, the results obtained from seven different many-objective optimization algorithms, viz. RD-EMO, NSGA-III, MOEA/D, MOEA/DD, RVEA, MOEA/D-M2M, and MaOJaya, are used to compare the performance (the abbreviations used in this paper and the full form are presented at the end of the paper).

4.1. Benchmark test problems

The performance of the proposed algorithm is investigated using the DTLZ (DTLZ1 to DTLZ5), a many-objective benchmark test function. A total of 20 benchmark functions with 3, 5, 8, and 10 objectives are considered. Each function has different features. The DTLZ1 function is linear and multi-modal. The DTLZ2 function is concave. The DTLZ3 function is concave and multi-modal. The DTLZ4 function is concave in type and biased, while the DTLZ5 function is concave and degenerate. These functions assess the performance of the proposed algorithms from different perspectives.

4.2. Parameter settings

The total number of design variables in the selected objective functions is calculated by using $(M + K - 1)$, where M represents the number of objectives and the value of K is 10, as discussed by Deb et al. [56].

The algorithms selected to design the proposed approach are randomized search techniques. The algorithm-specific parameters and population or swarm size, as well as the number of iterations to terminate the algorithms, need to be tuned. Initially, the selection of parameter values is chosen purely on a random basis. The value of these parameters is selected after executing the proposed approaches several times with different random values. Table 2 presents the parameter settings used for the proposed algorithms. The parameter settings for RD-EMO, NSGA-III, MOEA/D, MOEA/DD, RVEA, MOEA/D-M2M were taken from [57]. The proposed approach was executed on a system with an Intel Core i7 processor and Ubuntu 16.04LTS operating system. The simulation experiment of algorithms selected for comparison

Table 2. Parameter settings.

Test function	Design variables using M+K-1		Algorithm-specific and common control parameters		
	M	K	NS-MOPSO	NS-MODE	MaHODE
DTLZ1	3,5,8,10	5	• Weight factor = 0.4 • $C1$ and $C2 = 1$ • Population size = 200 • No. of iterations = 500	• Mutation rate = 0.4 • $F \in [0, 2] = 0.9$ • Population size = 200 • No. of iterations = 500	• Weight factor = 0.4 • $C1$ and $C2 = 1$ • Mutation rate = 0.4 • $F \in [0, 2] = 0.9$ • Population size = 200 • No. of iterations=1000 (Phase-I + Phase-II)
DTLZ2	3,5,8,10	10			
DTLZ3	3,5,8,10	10			
DTLZ4	3,5,8,10	10			
DTLZ5	3,5,8,10	10			

Note: $C1$: Personal coefficient; $C2$: Social coefficient; and F: Real and constant parameter.

purpose is performed using PlatEMO_v2.9.0 tool [58]. Each algorithm is executed independently 20 times.

4.2.1. Algorithm termination criterion:
The execution process of the algorithm ends when it reaches the maximum number of predefined iterations. The number of iterations used for the proposed approaches is purely selected after performing extensive experimentation. The proposed approaches terminate when they reach the pre-defined number of iterations.

4.3. Performance indicator

The performance of the proposed methods is investigated using 'IGD' and 'HV'. Also, another performance metric, used by Yang et al. [59], is average performance score, which is considered here. IGD measures the convergence and diversity of the obtained results. When the value of IGD is zero, it means that all the elements of the non-dominated set are in the Pareto-front. IGD measures the average Euclidean distance of the non-dominated element to the Pareto-front. IGD is mathematically represented in Eq. (2) and is taken from [1,60,61]:

$$IGD = \frac{\sqrt{\sum_{i=1}^{Q} di^2}}{Q}. \tag{2}$$

In Eq. (2), Q is the total sample points on PF and d_i is Euclidean distance between the obtained solution and sample points on the PF.

Another widely used performance indicator is HV which measures the closeness and diversity among the solutions; however, it is computationally expensive. It is computed using Eq. (3), taken from [60]:

$$HV = \text{volume}\left(\bigcup_{i=1}^{|Q|} v_i\right). \tag{3}$$

The obtained HV value is so biased that it has been eliminated by obtaining the HVR, which is the ratio of the obtained Pareto front (Q) to the best known Pareto front (P^*), as represented in Eq. (4). It is also taken from [60]:

$$HVR = \frac{HV(Q)}{HV(P*)}. \tag{4}$$

When all the objectives in MaOPs are of minimization type, the largest value of HVR is one.

Another performance metric used is 'average performance score'. It was described by Yang et al. [59] to measure the overall average performance of the proposed approach on the selected benchmark test functions. To compare the proposed performance algorithm (Ai), 'n' number of algorithms ($A1, A2, A3, \cdots, An$) are used. The proposed algorithm (Ai) obtains a better IGD value than any other algorithm used for comparison purpose; then, the V_{ij} is set to 1; otherwise, it is set to 0. The performance score for each algorithm is computed using Eq. (5) [59]. The minimum value of the average performance metric indicates the best results.

$$P(Ai) = \sum_{j=1, j\neq i}^{n} Vij. \tag{5}$$

Table 3 presents the comparison between the best values obtained using the proposed approaches and the algorithms selected from the literature for the DTLZ1 to DTLZ5 test functions for 3, 5, 8, and 10 objectives. The best results obtained are presented in boldface.

The results presented in Table 3 show the best IGD values obtained by MaOJaya, RD-EMO, NSGA-III, MOEA/D, MOEA/DD, RVEA, MOEA/D-M2M, NS-MODE, NS-MOPSO, and MaOHDE algorithms on DTLZ1 to DTLZ5 test functions for 3, 5, 8, and 10 objectives. After the comparison, it is observed that the MaOHDE algorithm outperforms other algorithms for most of the selected many-objective benchmark functions. The MaOHDE algorithm gives significantly better results for DTLZ1 and DTLZ3 test functions for

Table 3. Performance comparison of best Inverted Generational Distance (IGD) values obtained by Many-Objective Hybrid Differential Evolution (MaOHDE) and the other algorithms for DTLZ1 to DTLZ5 test functions for 3, 5, 8, and 10 objectives.

Func.	M	MaOJaya	RD-EMO	NSGA-III	MOEA/D	MOEA/DD	RVEA	MOEA/D-M2M	NS-MODE	NS-MOPSO	MaOHDE
DTLZ1	3	0.00E+00	2.95E-04	8.78E-04	4.17E-04	3.65E-04	5.54E-04	6.51E-02	4.56E-02	6.56E-02	**0.00E+00**
	5	**3.00E-06**	1.50E-04	6.52E-04	5.57E-04	2.72E-04	5.99E-04	2.98E-01	3.64E-03	5.16E-03	2.63E-05
	8	4.56E-03	1.56E-03	2.23E-03	3.96E-03	1.56E-03	4.49E-03	2.88E-01	4.33E-03	5.29E-03	**3.22E-04**
	10	4.17E-03	1.46E-03	2.45E-03	2.46E-03	1.91E-03	3.76E-03	1.59E-01	6.74E-03	6.87E-03	**3.72E-04**
DTLZ2	3	1.00E-03	3.80E-04	1.16E-03	5.49E-04	6.09E-04	1.75E-03	2.48E-02	2.39E-04	**2.03E-04**	2.40E-04
	5	7.72E-04	7.15E-04	3.86E-03	2.36E-03	1.29E-03	3.79E-03	2.18E-01	**6.38E-04**	6.18E-03	7.00E-04
	8	3.03E-03	1.30E-03	1.37E-02	2.72E-03	2.99E-03	1.41E-02	2.66E-01	5.11E-04	2.83E-03	**1.30E-04**
	10	2.02E-03	1.52E-03	1.74E-02	2.53E-03	3.36E-03	1.15E-02	2.65E-01	8.87E-03	1.33E-03	**3.21E-04**
DTLZ3	3	5.52E-03	3.69E-04	1.14E-03	1.25E-03	1.70E-03	3.56E-03	3.17E-01	8.03E-04	1.60E-02	**1.77E-04**
	5	4.65E-03	**2.97E-04**	2.62E-03	1.20E-03	6.45E-04	2.69E-03	1.98E+00	1.23E-03	1.98E-02	4.23E-04
	8	6.25E-04	2.16E-03	1.39E-02	9.62E-03	4.61E-03	1.69E-02	4.98E+00	1.07E-03	4.40E-02	**6.20E-04**
	10	8.06E-04	1.75E-03	9.76E-03	2.26E-03	2.91E-03	9.21E-03	6.27E-01	5.37E-04	5.61E-03	**2.57E-04**
DTLZ4	3	2.55E-03	8.28E-05	2.26E-04	1.99E-04	**1.17E-05**	4.88E-04	9.42E-03	1.12E-03	1.12E-03	1.89E-05
	5	4.06E-04	6.15E-05	4.70E-04	1.14E-04	9.68E-05	1.44E-03	**2.28E-05**	1.27E-03	1.89E-03	2.49E-05
	8	2.62E-04	7.49E-04	2.98E-03	2.25E-01	4.67E-04	7.03E-04	1.32E-01	**2.90E-04**	2.83E-03	7.15E-04
	10	2.76E-04	9.15E-04	3.95E-03	2.28E-03	1.40E-03	6.31E-03	1.45E-01	8.72E-03	2.00E-03	**1.05E-04**
DTLZ5	3	4.05E-02	2.32E-03	1.43E-02	3.05E-02	3.02E-02	6.63E-02	3.69E-02	3.64E-01	6.67E-01	**7.98E-04**
	5	2.09E-02	3.05E-02	**1.13E-03**	3.15E-02	1.07E-01	1.33E-01	3.05E-02	1.27E-02	1.98E-02	3.05E-02
	8	5.27E-02	**2.21E-02**	1.30E-01	2.39E-02	2.61E-02	2.36E-02	1.71E-01	3.51E-01	4.40E-02	2.64E-02
	10	1.16E-02	3.22E-02	2.29E-01	2.71E-02	1.01E-01	1.88E-01	2.00E-01	3.47E-01	6.38E-01	**1.09E-02**
+/−/=		3/15/2	2/14/4	1/19/0	1/19/0	3/17/0	2/18/0	1/18/1	3/17/0	1/18/1	

The values in boldface indicate better results.

Table 4. Statistic results of Inverted Generational Distance (IGD) comparison between Many-Objective Hybrid Differential Evolution (MaOHDE) and other multi/many-objective optimization evolutionary algorithms on DTLZ1 to DTLZ5 test functions.

Parameter	MaOJaya	RD-EMO	NSGA-III	MOEA/D	MOEA/DD	RVEA	MOEA/D-M2M	NS-MODE	NS-MOPSO	MaOHDE
Rank first	1	2	1	0	1	0	1	2	1	12
Better (+)	3	2	1	1	3	2	1	3	1	
Same (=)	2	4	0	0	0	0	1	0	1	
Worse (–)	15	14	19	19	17	18	18	17	18	

3, 8, and 10 objectives. Though the MaOHDE algorithm functions based on a non-dominated sorting approach, its performance is better than MOEA/DD and NSGA-III algorithms, as these algorithms make use of decomposition and non-dominated sorting approach, respectively. Out of other algorithms selected for comparison, the NS-MODE for DTLZ2 (5 objectives) and DTLZ4 (8 objectives) and RD-EMO for DTLZ3 (5 objectives) and DTLZ5 (10 objectives) algorithms give the best results. For the DTLZ2 test function, the MaOHDE algorithm outperforms MaOJaya, RD-EMO, NSGA-III, MOEA/D, MOEA/DD, RVEA, MOEA/D-M2M, and NS-MOPSO algorithms for all the selected objectives. However, the NS-MODE gives better values for the five objectives. The MaOHDE algorithm performs better for 3, 8, and 10 objectives of the DTLZ4 test function than MaOJaya, RD-EMO, NSGA-III, and MOEA/D algorithms. The MaOHDE obtains better values for the DTLZ5 test function for 3, 8, and 10 objectives than MaOJaya, NSGA-III, MOEA/D, MOEA/DD, RVEA, MOEA/D-M2M, NS-MODE, and NS-MOPSO algorithms. The proposed MaOHDE functions based on non-dominated sorting and decomposition-based approach and obtains better IGD values than NSGA-III (a non-dominated sorting based approach) and MOEA/D (a decomposition-based approach) for 90% of the selected test instances.

Table 4 presents the statistical results obtained

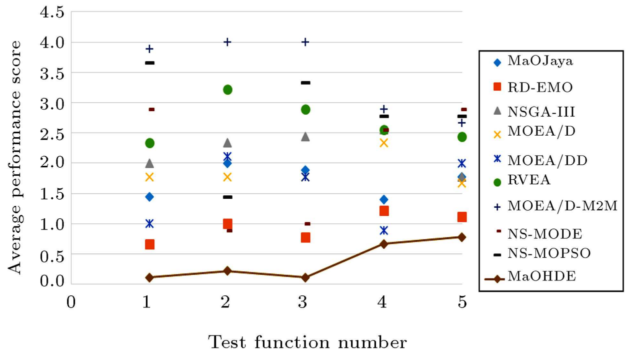


Figure 1. Average Inverted Generational Distance (IGD) performance scores for DTLZ test functions for different objectives.

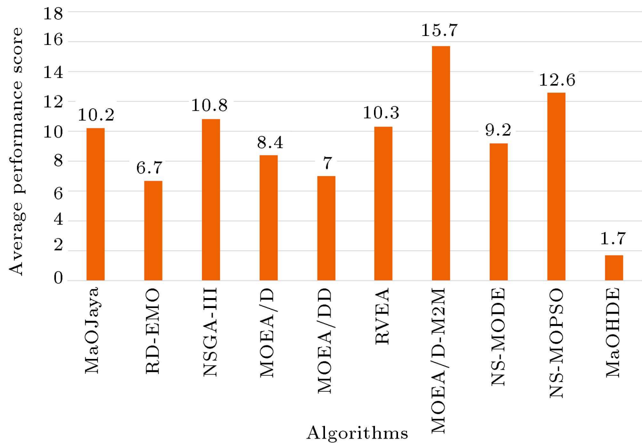


Figure 2. Average Inverted Generational Distance (IGD) performance scores of all algorithms on DTLZ Test functions for different objectives.

using Wilcoxon signed-rank method. The test is carried out at a significance level of 0.05. The +, −, and = symbols indicate whether the results obtained by the algorithms selected from the literature are significantly better, worse, and statistically similar to the results obtained by MaOHDE algorithms, respectively. The MaOHDE algorithm obtains the best value, i.e., first rank for 12 test functions out of 20. The RD-EMO and NS-MODE algorithms obtained the first rank for two test functions. The MaOJaya, NSGA-III, MOEA/DD, MOEA/D-M2M, and NS-MOPSO obtained the first rank for one time. The NS-MODE algorithm obtains the first rank for two test cases, while it obtains statistically better results than the MaOHDE algorithm for three test instances. The NS-MOPSO algorithm obtains the first rank for one test instance, while it gives statistically better and similar results for one test instance as compared to the MaOHDE algorithm. The MaOHDE succeeds for 60% of the test instances to obtain better results than other algorithms.

Figure 1 presents the scatter plots drawn based on the average performance score of MaOHDE and the other nine algorithms. It is computed using Eq. (5) on DTLZ1 to DTLZ5 test functions for the selected objectives.

Figure 1 shows that the average performance of the MaOHDE algorithm is significantly better than those of other algorithms. The MOEA/D-M2M performs the worst for all test functions in the case of the selected objectives. For the DTLZ5 test function, the performance of the MOEA/DD algorithm is close to that of the MaOHDE algorithm. The performance of the NS-MODE algorithm is close to the RD-EMO algorithm for DTLZ2 and DTLZ3 test functions. The RD-EMO algorithm's performance is close to the MaOHDE algorithm and it performs better than the other algorithms selected from the literature.

Figure 2 presents the histogram drawn by computing average performance scores for all the algorithms based on IGD values for DTLZ1 to DTLZ5 test functions on 3, 5, 8, and 10 objectives using Eq. (5).

From Figure 2, it is observed that the MaOHDE performs better than the other nine algorithms. The average performance of the NS-MODE algorithm is better than that of the NS-MOPSO algorithm. The RD-EMO, MOEA/D, and MOEA/DD algorithms perform better than the NS-MODE algorithm. The average performance index of NS-MODE is better than that of the NSGA-III algorithm. Figure 2 also confirms that the performance of the RD-EMO algorithm is closer to the MaOHDE algorithm and MOEA/D-M2M is inferior to all other algorithms.

Figure 3 presents a comparison between median IGD values obtained by the MaoHDE algorithm and those obtained by other nine different algorithms for selected DTLZ functions in the form of histograms. A total of 20 cases are presented in five plots. Each plot presents median IGD values obtained by ten different algorithms for one function with 3, 5, 8, and 10 objectives. From Figure 3, it is observed that the median values obtained by MaOHDE are not promising. In only a few cases, the MaOHDE obtains better median values. Therefore, there is a scope to improve the performance of the MaOHDE algorithm in terms of median values.

The best, mean, and worst values obtained by the MaOHDE and other nine algorithms are presented in the Appendix (Table A.1). Figure 4 presents the plot of best, mean, and worst values obtained by ten different algorithms on the DTLZ1 test function for 3, 5, 8, and 10 objectives. From Figure 4, it is observed that the MaOHDE algorithm obtains better IGD values for 3, 5, 8, and 10 objectives of a DTLZ1 test function than the other algorithms, selected for the study. The results presented in Table A.1 show that the MaOHDE algorithm performs better than the other nine algorithms selected for comparison. The MaOHDE performs better than MOEA/D and MOEA/D-M2M for all instances. These are decomposition-based

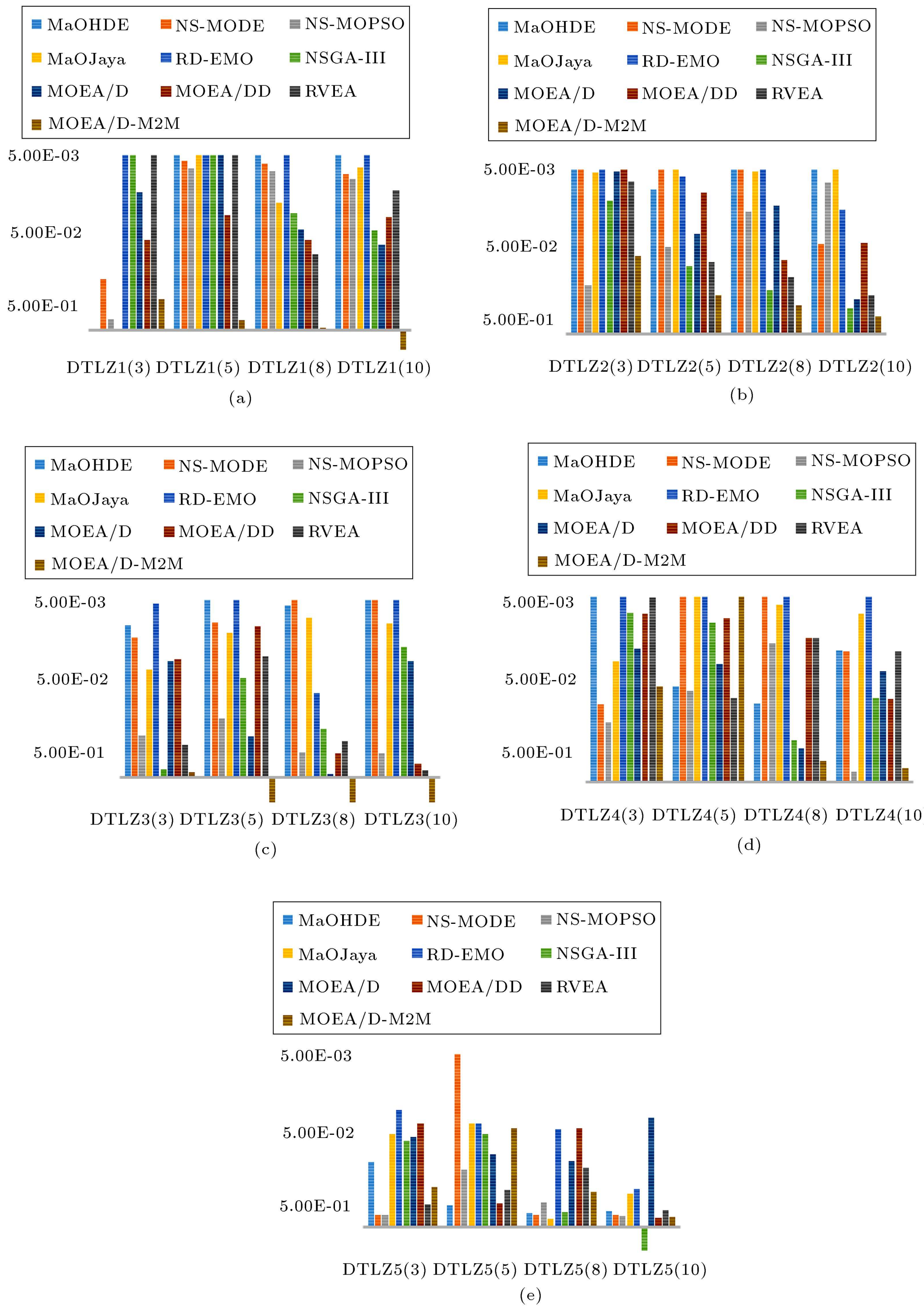
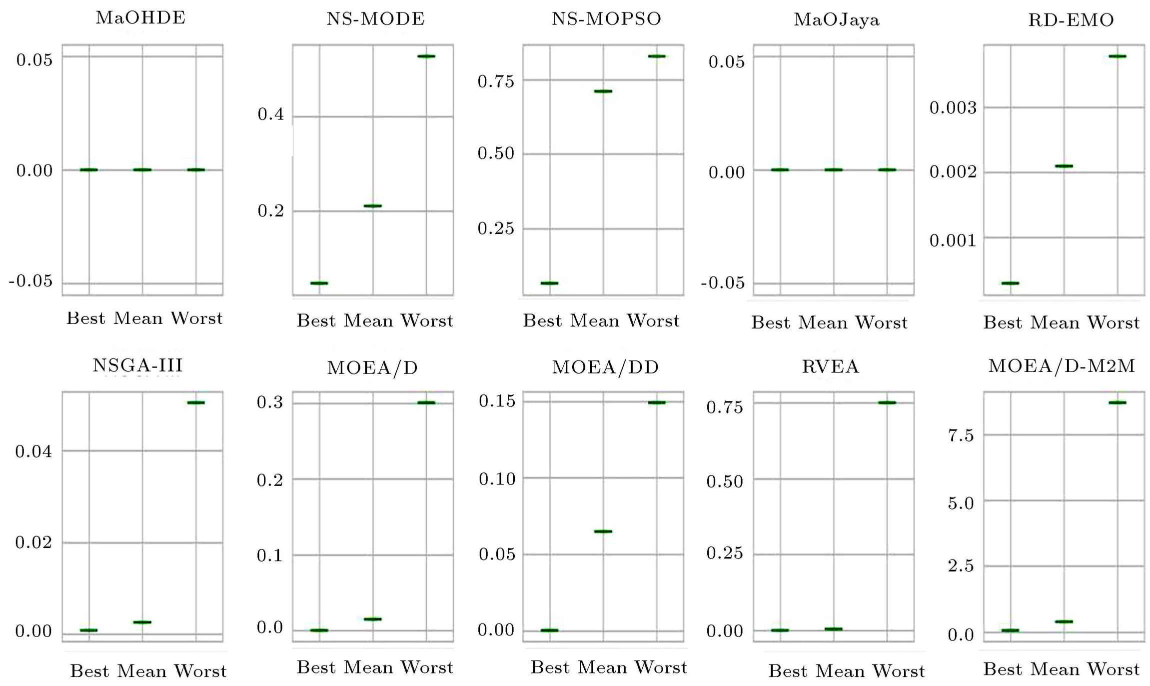


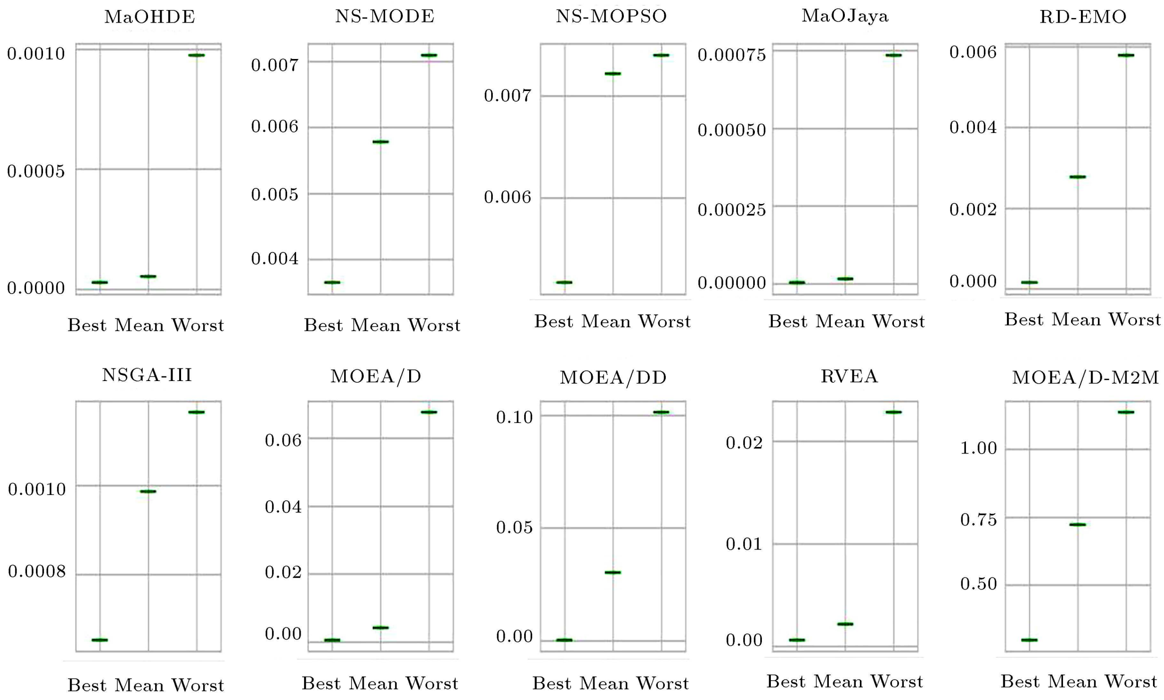
Figure 3. Comparison of median Inverted Generational Distance (IGD) values obtained by ten different algorithms for the selected benchmark functions: (a) DTLLZ1, (b) DTLZ2, (c) DTLLZ3, (d) DTLZ4, and (e) DTLZ5.

algorithms. Moreover, the performance of MaOHDE is comparatively better than the NSGA-III algorithm. The RD-EMO algorithm performs better than the MaOHDE algorithm for a few instances.

Table 5 presents the comparison of MaOHDE and other multi/many-objective optimization evolutionary algorithms based on the mean and standard deviation of a HV performance indicator. The HV mean and standard deviation values are computed for DTLZ1 to DTLZ5 test functions for 3 objectives. According to



(a) DTLZ1 (3 objectives)

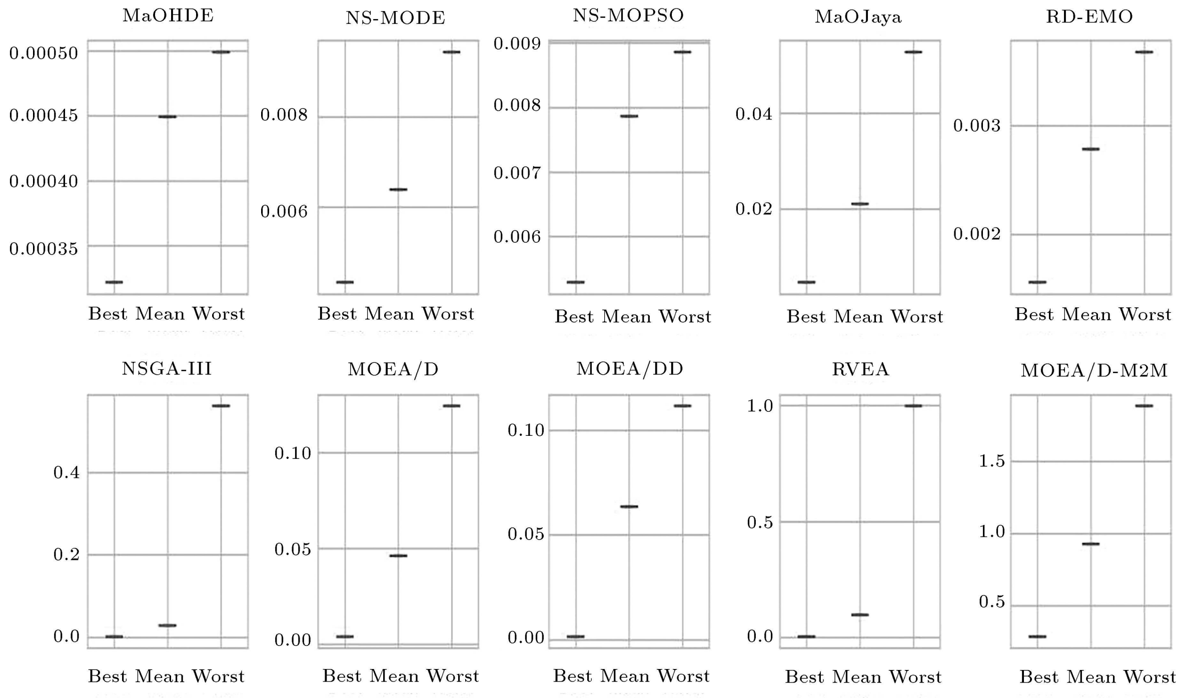


(b) DTLZ1 (5 objectives)

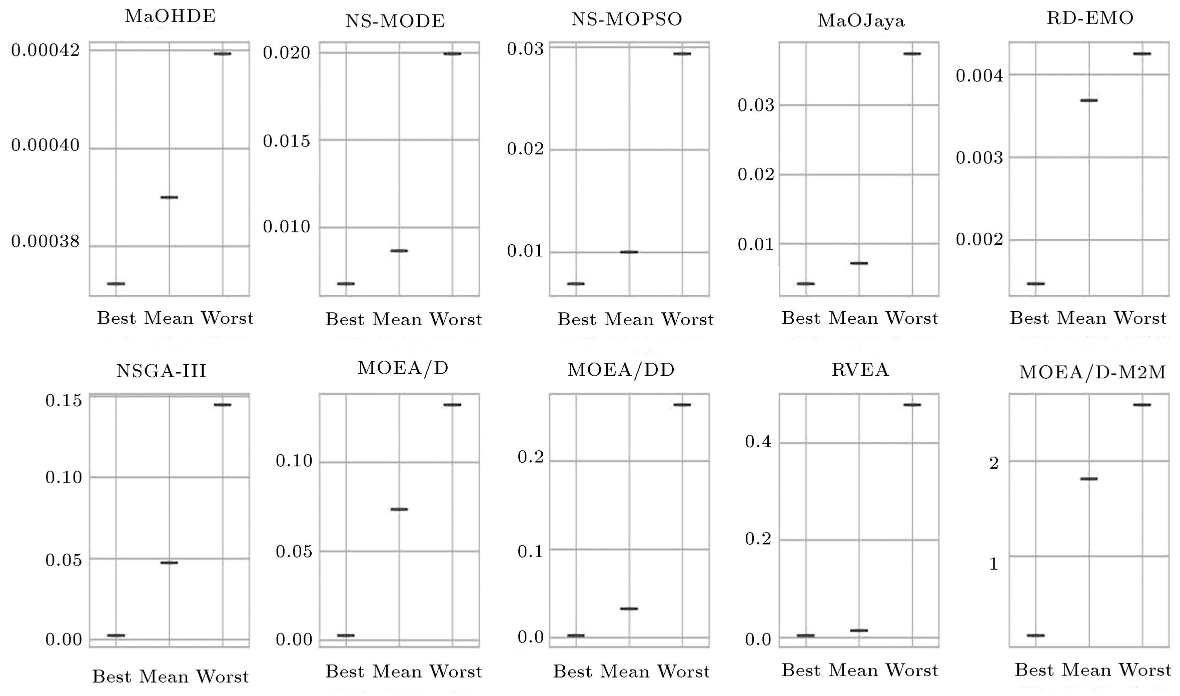
Figure 4. Comparison of best, mean, and worst Inverted Generational Distance (IGD) values obtained by ten different algorithms for DTLZ1 function for (a) 3 objectives, (b) 5 objectives, (c) 8 objectives, and (d) 10 objectives.

the computed results, it is observed that the MaOHDE performs better for DTLZ1, DTLZ2, and DTLZ4 test functions. The MaOJaya and NSGA-III obtain better results for DTLZ3 and DTLZ5 test functions, respectively. However, MaOHDE performs better than NS-MODE, NS-MOPSO, RD-EMO, MOEA/D, MOEA/DD, and MOEA/D-M2M algorithms for all the functions.

Figure 5 presents the convergence and diversity analysis of the proposed approach and its comparison



(c) DTLZ1 (8 objectives)



(d) DTLZ1 (10 objectives)

Figure 4. Comparison of best, mean, and worst Inverted Generational Distance (IGD) values obtained by ten different algorithms for DTLZ1 function for (a) 3 objectives, (b) 5 objectives, (c) 8 objectives, and (d) 10 objectives (continued).

with other selected multi/many-objective algorithms for DTLZ2 test function on 3 objectives. From Figure 5, it is observed that the MaOHDE algorithm converges faster than other algorithms. The performance of the RD-EMO algorithm is closer to the performance of the MaOHDE algorithm. These algorithms convergences at similar iterations. The NSGA-III and MOEA/DD-M2M algorithms require more iterations to converge than MaOHDE and NS-MODE algorithms.

Table 5. Comparison of Many-Objective Hybrid Differential Evolution (MaOHDE) and other multi/many-objective optimization evolutionary algorithms, hypervolume values obtained on DTLZ1 to DTLZ5 test functions for 3 objectives.

Func.	MaOHDE	NS-MODE	NS-MOPSO	MaOJaya	RD-EMO	NSGA-III	MOEA/D	MOEA/DD	RVEA	MOEA/D-M2M
DTLZ1	**9.9972E-01** (2.90E-05)	9.8001E-01 (9.21E-04)	9.8021E-01 (6.73E-03)	9.8013E-01 (5.60E-05)	9.9821E-01 (3.95E-03)	9.7493e-1 (3.31e-1)	8.7711e-1 (3.40e-1)	8.3684e-2 (1.24e-1)	9.7068e-1 (2.49e-1)	8.00020e+0 (2.35e-01)
DTLZ2	**9.9645E-01** (7.60E-05)	9.9626E-01 (2.01E-03)	9.2457E-01 (8.30E-03)	9.9545E-01 (7.51E-04)	9.7281E-01 (2.39E-04)	9.5550e-1 (5.88e-4)	8.5522e-1 (8.13e-4)	8.5428e-1 (1.01e-3)	9.5262e-1 (1.55e-3)	9.3609e-1 (1.58e-2)
DTLZ3	9.9601E-01 (1.24E-04)	9.7343E-01 (7.80E-03)	9.6870E-01 (3.32E-01)	**9.9743E-01** (2.23E-04)	9.971E-01 (1.29E-04)	9.2650e-1 (6.58e-4)	8.9265e-1 (4.68e-1)	8.5876e-1 (4.01e-2)	9.2672e-1 (1.55e-3)	9.2043e-01 (3.62e-02)
DTLZ4	**9.6702E-01** (1.98E-04)	9.1898E-01 (6.64E-03)	7.9655E-01 (6.09E-03)	9.6009E-01 (8.90E-04)	9.427E-01 (5.63E-03)	9.6621e-1 (1.34e-1)	8.7765e-1 (1.58e-1)	9.5449e-1 (1.05e-3)	9.5254e-1 (1.86e-3)	9.3339e-1 (1.79e-2)
DTLZ5	9.8436E-01 (1.56E-3)	9.3683E-01 (3.09E-3)	8.9908E-01 (5.61E-03)	9.8287E-01 (3.68E-03)	9.8421E-01 (2.36E-3)	**9.9285e-1** (1.73e-3)	8.8251e-1 (3.05e-4)	8.8249e-1 (7.44e-4)	8.4485e-1 (8.73e-3)	8.6376e-1 (7.56e-3)

The values in boldface indicate better results.

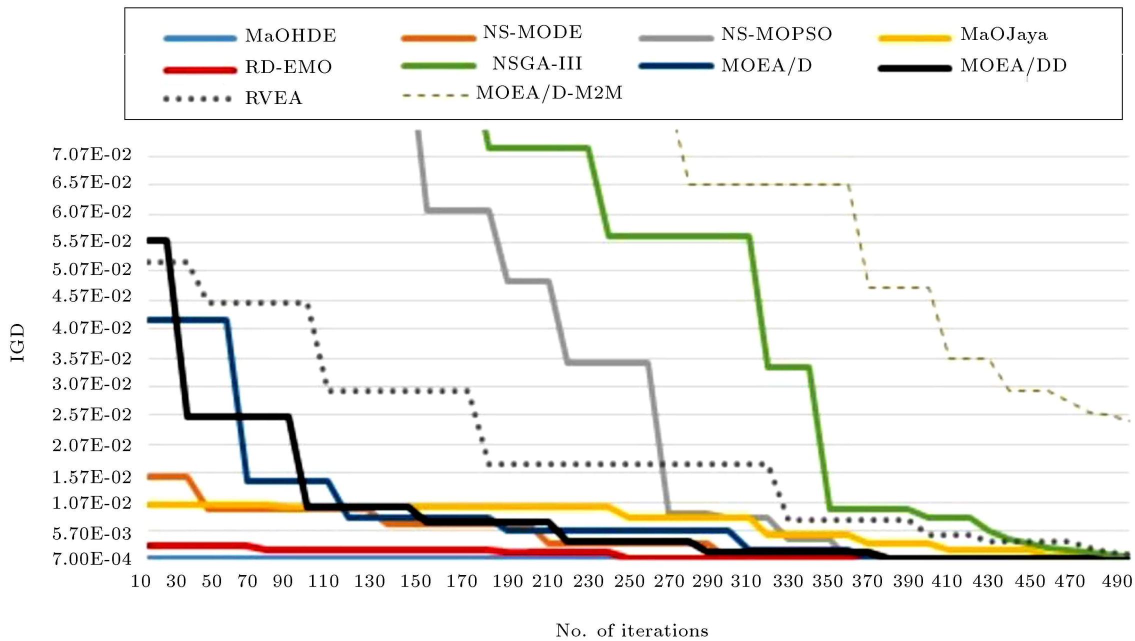


Figure 5. Convergence analysis for DTLZ2 Test function for 3 objectives in terms of Inverted Generational Distance (IGD).

5. Conclusions and future work

This paper presented Many-Objective Hybrid Differential Evolution (MaOHDE), a hybrid many-objective evolutionary algorithm, to address the Many-objective Optimization Problems (MaOPs). The Non-dominated Sorting based Multi-Objective Differential Evolution (NS-MODE) and Non-dominated Sorting based Multi-Objective Particle Swarm Optimization (NS-MOPSO) algorithms were designed to develop the MaOHDE algorithm. The proposed approaches integrated the 'non-dominated sorting approach from NSGA-II', the ranking scheme, reference points, and the weight vector. The Tchebycheff, a decomposition-based approach, was utilized to decompose the MaOPs into several single-objective optimization problems. The efficiency and effectiveness of the proposed approaches were evaluated with respect to DTLZ1 to DTLZ5 benchmark functions for 3, 5, 8, and 10 objectives. Also, the proposed approaches were compared with seven state-of-the-art algorithms, which are based on decomposition and a non-dominated sorting scheme. Various performance metrics and plots were used to present the comparative performance of the proposed approach. The results showed that the MaOHDE algorithm obtained the first rank for 60% of the test instances. However, the NS-MODE and NS-MOPSO, when applied individually to solve MaOPs, were not found effective. Also, the convergence rate of MaOHDE was better than other algorithms used for comparison. The results confirmed that a single algorithm could not be effective in addressing the MaOPs with diverse properties and objectives. Though the proposed approach found it effective to address the MaOPs, there exist few limitations. The proposed MaOHDE algorithm fails to obtain better results for 40% of the test instances. The hybrid approach combines NS-MODE with NS-MOPSO and it was found that the number of generations required to obtain the best results was two times that of NS-MODE and NS-MOPSO algorithms. The hybrid approach had the input from the NS-MOPSO algorithm's output; therefore, the difference among the obtained solutions was reduced.

As for the future work, the performance of the proposed approach can be improved in terms of diversity. The proposed approach can be applied to real-time as well as large-scale MaOPs to measure its effectiveness and efficiency.

Acknowledgement

The authors would like to thank the editorial team and anonymous reviewers for helping us to enhance the article's quality and contribution with their suggestions. Also, we would like to thank Mr. Abhijit Chavan for extending his kind help to carry out the experimentation.

Abbreviations

MaOPs	Many-objective Optimization Problems
MOPs	Multi-objective Optimization Problems
PF	Pareto Front
PS	Pareto Set
MOEAs	Multi-objective Evolutionary Algorithms
NS-MODE	Non-dominated Sorting based Multi-Objective Differential Evolution
NS-MOPSO	Non-dominated Sorting based Multi-Objective Particle Swarm Optimization
MaOHDE	Many-Objective Hybrid Differential Evolution
EMOAs	Evolutionary Multi-Objective Algorithms
NSGA-II	Non-dominated Sorting based Genetic Algorithm-II
NSGA-III	Non-dominated Sorting based Genetic Algorithm-III
MOEA/D	Multi-Objective Evolutionary Algorithm based on Decomposition
MOEA/DD	Multi-Objective Evolutionary Algorithm based on Dominance and Decomposition
REVA	Reference vector guided evolutionary algorithm
RD-EMO	Region division-based decomposition approach to evolutionary many-objective optimization
MaOJaya	Many-Objective Jaya
IGD	Inverted Generational Distance
HV	Hypervolume

References

1. Mohammadi, A., Omidvar, M.N., Li, X., et al. "Integrating user preferences and decomposition methods for many-objective optimization", *Proc. 2014 IEEE Congr. Evol. Comput. CEC 2014*, pp. 421–428 (2014).

2. Cheng, R., Jin, Y., Olhofer, M., and Sendhoff, B. "A reference vector guided evolutionary algorithm for many-objective optimization", *IEEE Trans. Evol. Comput.*, **20**(5), pp. 773–791 (2016).

3. Li, K., Deb, K., Zhang, Q., and Kwong, S. "An evolutionary many-objective optimization algorithm based on dominance and decomposition", *IEEE Trans. Evol. Comput.*, **19**(5), pp. 694–716 (2015).

4. Giagkiozis, I. and Fleming, P.J. "Methods for many-objective optimization: An analysis", **1030**, pp. 1–15 (2012).

5. Giagkiozis, I. and Fleming, P.J. "Methods for multi-objective optimization: An analysis", *Inf. Sci. (Ny).*, **293**, pp. 338–350 (2015).

6. Ma, X., Yang, J., Wu, N., et al. "A comparative study on decomposition-based multi-objective evolutionary algorithms for many-objective optimization", *2016 IEEE Congr. Evol. Comput. CEC 2016*, pp. 2477–2483 (2016).

7. Wang, H., Jin, Y., and Yao, X. "Diversity assessment in many-objective optimization", *IEEE Trans. Cybern.*, **47**(6), pp. 1510–1522 (2017).

8. Ishibuchi, H., Tsukamoto, N., and Nojima, Y. "Evolutionary many-objective optimization: A short review", *2008 IEEE Congr. Evol. Comput. CEC 2008*, pp. 2424–2431 (2008).

9. Marler, R.T. and Arora, J.S. "Survey of multi-objective optimization methods for engineering", *Struct. Multidiscip. Optim.*, **26**(6), pp. 369–395 (2004).

10. Batista, L.S., Campelo, F., Guimarães, F.G., et al. "A comparison of dominance criteria in many-objective optimization problems", *2011 IEEE Congr. Evol. Comput. CEC 2011*, pp. 2359–2366 (2011).

11. Trivedi, A., Srinivasan, D., Sanyal, K., et al. "A survey of multiobjective evolutionary algorithms based on decomposition", *IEEE Trans. Evol. Comput.*, **21**(3), pp. 440–462 (2016).

12. Mane, S.U. and Narasinga Rao, M.R. "Many-objective optimization: Problems and evolutionary algorithms - a short review", *Int. J. Appl. Eng. Res.*, **12**(20), pp. 9774–9793 (2017).

13. Arabas, K. and Opara, J. "Benchmarking procedures for continuous optimization algorithms", *J. Telecommun. Inf. Technol.*, **4**, pp. 73–80 (2011).

14. Jamil, M. and Yang, X.S. "A literature survey of benchmark functions for global optimisation problems", *Int. J. Math. Model. Numer. Optim.*, **4**(2), pp. 150–194 (2013).

15. Rao, S.S., *Engineering Optimization: Theory and Practice*, 5th Edn., John Wiley and Sons (2019).

16. Rao, R. "Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems", *Int. J. Ind. Eng. Comput.*, **7**(1), pp. 19–34 (2016).

17. Deb, K. and Jain, H. "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, Part I: Solving problems with box constraints", *IEEE Trans. Evol. Comput.*, **18**(4), pp. 577–601 (2014).

18. Dai, S., Liu, H., and Chen, L. "Evolutionary many-objective optimization algorithm based on improved K-dominance and M2M population decomposition",

Proc. - 2015 11th Int. Conf. Comput. Intell. Secur. CIS 2015, pp. 286–290 (2016).

19. Pal, M., Saha, S., and Bandyopadhyay, S. "Clustering based online automatic objective reduction to aid many-objective optimization", *2016 IEEE Congr. Evol. Comput. CEC 2016*, pp. 1131–1138 (2016).

20. Ishibuchi, H., Akedo, N., and Nojima, Y. "Behavior of multiobjective evolutionary algorithms on many-objective knapsack problems", *IEEE Trans. Evol. Comput.*, **19**(2), pp. 264–283 (2014).

21. Von Lücken, C., Barán, B., and Brizuela, C. "A survey on multi-objective evolutionary algorithms for many-objective problems", *Comput. Optim. Appl.*, **58**(3), pp. 707–756 (2014).

22. Nguyen, X.H., Bui, L.T., and Tran, C.T. "Improving many objective optimisation algorithms using objective dimensionality reduction", *Evol. Intell.*, **13**(3), pp. 365–380 (2020).

23. Zhou, A., Wang, Y., and Zhang, J. "Objective extraction via fuzzy clustering in evolutionary many-objective optimization", *Inf. Sci. (Ny).*, **509**, pp. 343–355 (2020).

24. Zhao, H. and Zhang, C. "An online-learning-based evolutionary many-objective algorithm", *Inf. Sci. (Ny).*, **509**(195), pp. 1–21 (2020).

25. Jiang, S., He, X., and Zhou, Y. "Many-objective evolutionary algorithm based on adaptive weighted decomposition", *Appl. Soft Comput. J.*, **84**, p. 105731 (2019).

26. Chen, H., Cheng, R., Pedrycz, W., et al. "Solving many-objective optimization problems via multistage evolutionary search", *IEEE Trans. Syst. Man, Cybern. Syst.*, pp. 2168–2216 (2019).

27. Li, F., Liu, J., Huang, P., et al. "An R2 indicator and decomposition based steady-state evolutionary algorithm for many-objective optimization", *Math. Probl. Eng.*, **2018**, pp. 1–18 (2018).

28. Wang, W.L., Li, W., and Wang, Y.L. "An opposition-based evolutionary algorithm for many-objective optimization with adaptive clustering mechanism", *Comput. Intell. Neurosci.* (2019).

29. Liu, Y., Qin, H., Zhang, Z., et al. "A region search evolutionary algorithm for many-objective optimization", *Inf. Sci. (Ny).*, **488**, pp. 19–40 (2019).

30. Li, K., Wang, R., Zhang, T., et al. "Evolutionary many-objective optimization: A comparative study of the state-of-the-art", *IEEE Access*, **6**, pp. 26194–26214 (2018).

31. Mane, S. and Narsingrao, M. "A chaotic-based improved many-objective jaya algorithm for many-objective optimization problems", *Int. J. Ind. Eng. Comput.*, **12**(1), pp. 49–62 (2021).

32. Bandyopadhyay, S. and Mukherjee, A. "An algorithm for many-objective optimization with reduced objective computations: A study in differential evolution", *IEEE Trans. Evol. Comput.*, **19**(3), pp. 400–413 (2015).

33. Britto, A. and Pozo, A. "Reference-point based multi-swarm algorithm for many-objective problems", *Proc. - 2015 Brazilian Conf. Intell. Syst. BRACIS 2015*, pp. 252–257 (2016).

34. Deb, K., Sundar, J., Udaya Bhaskara, R.N., et al. "Reference point based multi-objective optimization using evolutionary algorithms", *Int. J. Comput. Intell. Res.*, **2**(3), pp. 273–286 (2006).

35. Storn, R. and Price, K. "Differential evolution - a simple and efficient heuristic for global optimization over continuous space", *J. Glob. Optim.*, **11**(4), pp. 341–359 (1997).

36. Zhao, M., Liu, R., Li, W., et al. "Multi-objective optimization based differential evolution constrained optimization algorithm", *Proc. - 2010 2nd WRI Glob. Congr. Intell. Syst. GCIS 2010*, **1**(1), pp. 320–326 (2010).

37. Kachitvichyanukul, V. "Comparison of three evolutionary algorithms: GA, PSO, and DE", *Ind. Eng. Manag. Syst.*, **11**(3), pp. 215–223 (2012).

38. Das, S., Mullick, S.S., and Suganthan, P.N. "Recent advances in differential evolution-An updated survey", *Swarm Evol. Comput.*, **27**, pp. 1–30 (2016).

39. Thomas, Z., Poupart, P., Kennedy, J., et al. "Particle swarm optimization", *Encyclopedia of Machine Learning*, **1**(1), pp. 760–766 (2011).

40. Feng, S., Kang, Q., Chen, X., et al. "A weight-aggregation multi-objective PSO algorithm for load scheduling of PHEVs", *2016 IEEE Congr. Evol. Comput. CEC 2016*, pp. 2896–2902 (2016).

41. Song, M.P. and Gu, G.C. "Research on particle swarm optimization: A review", *Proc. 2004 Int. Conf. Mach. Learn. Cybern.*, **4**, pp. 2236–2241 (2004).

42. Rini, D.P., Shamsuddin, S.M., and Yuhaniz, S.S. "Particle swarm optimization: Technique, system and challenges", *Int. J. Appl. Inf. Syst.*, **1**(1), pp. 33–45 (2011).

43. Poli, R., Kennedy, J., and Blackwell, T. "Particle swarm optimization: An overview", *Swarm Intell.*, **1**(1), pp. 33–57 (2007).

44. Banks, A., Vincent, J., and Anyakoha, C. "A review of particle swarm optimization. Part I: Background and development", *Nat. Comput.*, **6**(4), pp. 467–484 (2007).

45. Banks, A., Vincent, J., and Anyakoha, C. "A review of particle swarm optimization. Part II: Hybridisation, combinatorial, multicriteria and constrained optimization, and indicative applications", *Nat. Comput.*, **7**(1), pp. 109–124 (2008).

46. Rana, S., Jasola, S., and Kumar, R. "A review on particle swarm optimization algorithms and their applications to data clustering", *Artif. Intell. Rev.*, **35**(3), pp. 211–222 (2011).

47. Garg, H. and Sharma, S.P. "Multi-objective reliability-redundancy allocation problem using particle swarm optimization", *Comput. Ind. Eng.*, **64**(1), pp. 247–255 (2013).

48. Dragoi, E.N. and Dafinescu, V. "Parameter control and hybridization techniques in differential evolution: a survey", *Artif. Intell. Rev.*, **45**(4), pp. 447–470 (2016).

49. Jitkongchuen, D. "A hybrid differential evolution with grey Wolf optimizer for continuous global optimization", *Proc. - 2015 7th Int. Conf. Inf. Technol. Electr. Eng. Envisioning Trend Comput. Inf. Eng. ICITEE 2015*, pp. 51–54 (2015).

50. Liu, T. and Maeda, M. "Set-based differential evolution for traveling salesman problem", *Proc. - 2013 6th Int. Conf. Intell. Networks Intell. Syst. ICINIS 2013*, pp. 107–110 (2013).

51. Qin, Y., Hu, H., Shi, Y., et al. "An artificial bee colony algorithm hybrid with differential evolution for multi-temporal image registration", *35th Chinese Control Conf. CCC*, **2016**, pp. 2734–2739 (2016).

52. Ye, S., Dai, G., Peng, L., et al. "A hybrid adaptive co-evolutionary differential evolution algorithm for large-scale optimization", *Proc. 2014 IEEE Congr. Evol. Comput. CEC 2014*, pp. 1277–1284 (2014).

53. Garg, H. "A hybrid GA-GSA algorithm for optimizing the performance of an industrial system by utilizing uncertain data", In *Handbook of Research on Artificial Intelligence Techniques and Algorithms*, pp. 620–654 (2015).

54. Garg, H. "A hybrid PSO-GA algorithm for constrained optimization problems", *Appl. Math. Comput.*, **274**, pp. 292–305 (2016).

55. Patwal, R.S., Narang, N., and Garg, H. "A novel TVAC-PSO based mutation strategies algorithm for generation scheduling of pumped storage hydrothermal system incorporating solar units", *Energy*, **142**, pp. 822–837 (2018).

56. Deb, K., Thiele, L., Laumanns, M., et al. "Scalable test problems for evolutionary multiobjective optimization", *Evol. Multiobjective Optim.*, **1990**, pp. 105–145 (2005).

57. Liu, R., Liu, J., Zhou, R., et al. "A region division based decomposition approach for evolutionary many-objective optimization", *Knowledge-Based Syst.*, **194**, pp. 105–518 (2020).

58. Tian, Y., Cheng, R., Zhang, X., et al. "PlatEMO: A MATLAB platform for evolutionary multi-objective optimization [educational forum]", *IEEE Comput. Intell. Mag.*, **12**(4), pp. 73–87 (2017).

59. Yang, W., Chen, L., Wang, Y., et al. "A reference points and intuitionistic fuzzy dominance based particle swarm algorithm for multi/many-objective optimization", *Appl. Intell.*, **50**(4), pp. 1133–1154 (2020).

60. Lwin, K., Qu, R., and Kendall, G. "A learning-guided multi-objective evolutionary algorithm for constrained portfolio optimization", *Appl. Soft Comput. J.*, **24**, pp. 757–772 (2014).

61. Radziukynienė, I. and Žilinskas, A. "Evolutionary methods for multi-objective portfolio optimization", *Lect. Notes Eng. Comput. Sci.*, **2171**(1), pp. 1155–1159 (2008).

Appendix

In Table A.1, it is presented that the MaOHDE algorithm performs better than the other nine algorithms selected for comparison.

Table A.1. Comparison of best, mean, and worst Inverted Generational Distance (IGD) values obtained by Many-Objective Hybrid Differential Evolution (MaOHDE) and other multi/many-objective optimization evolutionary algorithms on DTLZ1 to DTLZ5 test functions.

Func.	*M*	Test	MaOHDE	NS-MODE	NS-MOPSO	MaOJaya	RD-EMO	NSGA-III	MOEA/D	MOEA/DD	RVEA	MOEA/D-M2M
DTLZ1	3	Best	**0.00E+00**	4.56E-02	6.56E-02	0.00E+00	2.95E-04	8.78E-04	4.17E-04	3.65E-04	5.54E-04	6.51E-02
		Mean	**0.00E+00**	2.10E-01	7.10E-01	0.00E+00	2.09E-03	2.57E-03	1.52E-02	6.48E-02	3.66E-03	3.89E-01
		Worst	**0.00E+00**	5.28E-01	8.28E-01	0.00E+00	3.78E-03	5.03E-02	3.01E-01	1.49E-01	7.49E-01	8.69E+00
	5	Best	2.63E-05	3.64E-03	5.16E-03	**3.00E-06**	1.50E-04	6.52E-04	5.57E-04	2.72E-04	5.99E-04	2.98E-01
		Mean	5.15E-05	5.78E-03	7.22E-03	**1.55E-05**	2.77E-03	9.87E-04	4.05E-03	3.01E-02	2.12E-03	7.22E-01
		Worst	9.75E-04	7.10E-03	7.40E-03	**7.35E-04**	5.79E-03	1.16E-03	6.75E-02	1.01E-01	2.28E-02	1.14E+00
	8	Best	**3.22E-04**	4.33E-03	5.29E-03	4.56E-03	1.56E-03	2.23E-03	3.96E-03	1.56E-03	4.49E-03	2.88E-01
		Mean	**4.49E-04**	6.39E-03	7.87E-03	2.10E-02	2.78E-03	2.84E-02	4.61E-02	6.36E-02	9.76E-02	9.27E-01
		Worst	**4.99E-04**	9.44E-03	8.86E-03	5.28E-02	3.67E-03	5.60E-01	1.24E-01	1.11E-01	9.95E-01	1.88E+00
	10	Best	**3.72E-04**	6.74E-03	6.87E-03	4.17E-03	1.46E-03	2.45E-03	2.46E-03	1.91E-03	3.76E-03	1.59E-01
		Mean	**3.90E-04**	8.65E-03	9.96E-03	7.15E-03	3.68E-03	4.73E-02	7.36E-02	3.21E-02	1.42E-02	1.81E+00
		Worst	**4.19E-04**	1.99E-02	2.94E-02	3.74E-02	4.25E-03	1.44E-01	1.32E-01	2.63E-01	4.77E-01	2.59E+00
DTLZ2	3	Best	2.40E-04	3.92E-04	**2.03E-04**	1.00E-03	3.80E-04	1.16E-03	5.49E-04	6.09E-04	1.75E-03	2.48E-02
		Mean	**4.22E-04**	3.66E-03	2.06E-03	5.28E-03	2.67E-03	1.31E-02	5.12E-03	3.71E-03	7.12E-03	7.90E-02
		Worst	**1.02E-03**	1.51E-02	6.47E-01	1.01E-02	3.07E-03	6.21E-01	4.24E-02	5.59E-02	5.24E-02	1.56E-01
	5	Best	7.00E-04	**6.38E-04**	6.18E-03	7.72E-04	7.15E-04	3.86E-03	2.36E-03	1.29E-03	3.79E-03	2.18E-01
		Mean	9.25E-03	2.99E-03	6.03E-02	**2.58E-03**	6.03E-03	1.11E-01	3.92E-02	1.03E-02	9.71E-02	2.81E-01
		Worst	2.59E-02	6.56E-03	3.19E-01	**4.19E-03**	6.58E-03	2.18E-01	1.96E-01	2.14E-01	1.98E-01	4.83E-01
	8	Best	**1.30E-04**	5.11E-04	2.83E-03	3.03E-03	1.30E-03	1.37E-02	2.72E-03	2.99E-03	1.41E-02	2.66E-01
		Mean	**5.18E-04**	2.39E-03	1.88E-02	5.17E-03	2.31E-03	2.41E-01	1.56E-02	8.92E-02	1.58E-01	3.87E-01
		Worst	2.09E-03	**1.45E-03**	2.42E-01	4.19E-03	2.93E-03	4.15E-01	3.27E-01	3.92E-01	3.34E-01	8.40E-01
	10	Best	**3.21E-04**	8.87E-03	1.33E-03	2.02E-03	1.52E-03	1.74E-02	2.53E-03	3.36E-03	1.15E-02	2.65E-01
		Mean	**3.90E-03**	5.38E-02	7.52E-03	4.50E-03	1.81E-02	4.33E-01	3.23E-01	5.22E-02	2.82E-01	5.62E-01
		Worst	1.37E-02	2.17E-01	1.86E-01	**6.09E-03**	2.01E-02	6.50E-01	4.44E-01	5.01E-01	4.45E-01	1.04E+00
DTLZ3	3	Best	**1.77E-04**	8.03E-04	1.60E-02	5.52E-03	3.69E-04	1.14E-03	1.25E-03	1.70E-03	3.56E-03	3.17E-01
		Mean	**1.06E-03**	1.53E-02	2.82E-01	3.95E-02	5.39E-03	7.96E-01	3.02E-02	2.91E-02	3.77E-01	8.62E-01
		Worst	7.88E-02	4.12E-02	1.25E+00	1.02E-01	**6.82E-03**	1.36E+00	6.02E-01	2.56E-01	7.55E+00	1.05E+00
	5	Best	4.23E-04	1.23E-03	1.98E-02	4.65E-03	**2.97E-04**	2.62E-03	1.20E-03	6.45E-04	2.69E-03	1.98E+00
		Mean	**4.90E-04**	9.70E-03	1.72E-01	1.31E-02	2.87E-03	5.03E-02	2.91E-01	1.08E-02	2.67E-02	5.78E+00
		Worst	**4.24E-03**	1.46E-02	9.57E-01	1.93E-02	4.87E-03	4.97E-01	3.79E-01	1.94E-02	3.29E-01	9.53E+00
	8	Best	**6.20E-04**	1.07E-03	4.40E-02	6.25E-04	2.16E-03	1.39E-02	9.62E-03	4.61E-03	1.69E-02	4.98E+00
		Mean	5.73E-03	**2.13E-03**	4.77E-01	8.31E-03	7.91E-02	2.30E-01	9.02E-01	4.91E-01	3.41E-01	5.61E+00
		Worst	1.88E-02	5.19E-03	2.79E-01	**2.88E-03**	9.75E-02	6.38E+00	4.16E+00	2.97E+00	2.84E+00	1.05E+01
	10	Best	**2.57E-04**	5.37E-04	5.61E-03	8.06E-04	1.75E-03	9.76E-03	2.26E-03	2.91E-03	9.21E-03	6.27E-01
		Mean	**6.27E-04**	1.76E-03	4.80E-01	9.99E-03	2.01E-03	2.02E-02	3.07E-02	6.59E-01	8.02E-01	5.96E+00
		Worst	**2.15E-03**	3.44E-03	5.07E+00	2.93E-03	2.22E-03	3.94E-01	2.10E-01	1.19E+00	1.15E+00	6.56E+00

The values in boldface indicate better results.

Table A.1. Comparison of best, mean, and worst Inverted Generational Distance (IGD) values obtained by Many-Objective Hybrid Differential Evolution (MaOHDE) and other multi/many-objective optimization evolutionary algorithms on DTLZ1 to DTLZ5 test functions (continued).

Func.	*M*	Test	MaOHDE	NS-MODE	NS-MOPSO	MaOJaya	RD-EMO	NSGA-III	MOEA/D	MOEA/DD	RVEA	MOEA/D-M2M
DTLZ4	3	Best	1.89E-05	1.12E-02	1.12E-02	2.55E-03	8.28E-05	2.26E-04	1.99E-04	**1.17E-05**	4.88E-04	9.42E-03
		Mean	4.31E-04	1.06E-01	1.81E-01	3.11E-02	**1.43E-04**	7.72E-03	2.16E-02	7.92E-03	5.03E-03	6.41E-02
		Worst	1.01E-02	7.88E-01	5.69E-01	8.06E-02	**1.85E-04**	5.51E-02	1.24E-01	5.57E-02	4.94E-02	1.26E-01
	5	Best	2.49E-05	1.27E-03	1.89E-03	4.06E-04	6.15E-05	4.70E-04	1.14E-04	9.68E-05	1.44E-03	**2.28E-05**
		Mean	6.42E-02	2.52E-03	7.20E-02	1.24E-03	**5.67E-04**	1.02E-02	3.32E-02	9.01E-03	8.91E-02	4.24E-03
		Worst	4.28E-01	5.60E-03	1.65E+00	2.19E-03	**9.78E-04**	2.19E-01	6.46E-01	2.16E-02	1.98E-01	4.87E-03
	8	Best	7.15E-04	2.90E-04	2.83E-03	**2.62E-04**	7.49E-04	2.98E-03	2.25E-01	4.67E-04	7.03E-04	1.32E-01
		Mean	**1.03E-03**	2.48E-03	1.88E-02	6.08E-03	3.67E-03	3.02E-01	3.81E-01	1.61E-02	1.61E-02	5.41E-01
		Worst	4.26E-01	**5.70E-03**	2.42E-01	3.61E-02	**5.70E-03**	4.00E-01	1.05E+00	3.45E-02	3.46E-02	8.66E-01
	10	Best	**1.05E-04**	8.72E-03	2.00E-03	2.76E-04	9.15E-04	3.95E-03	2.28E-03	1.40E-03	6.31E-03	1.45E-01
		Mean	2.27E-03	2.34E-02	7.42E-01	7.88E-03	**1.32E-03**	9.01E-02	4.14E-02	9.22E-02	2.32E-02	6.58E-01
		Worst	2.60E-02	3.92E-02	3.61E-01	1.02E-01	**1.64E-03**	5.76E-01	7.59E-01	4.57E-01	4.60E-01	1.03E+00
DTLZ5	3	Best	**7.98E-04**	3.64E-01	6.67E-01	4.05E-02	2.32E-03	1.43E-02	3.05E-02	3.02E-02	6.63E-02	3.69E-02
		Mean	**1.34E-03**	6.88E-01	6.92E-01	5.70E-02	2.76E-02	6.97E-02	6.17E-02	4.16E-02	5.04E-01	2.91E-01
		Worst	4.17E-01	7.30E-01	7.39E-01	1.04E-01	4.31E-02	1.37E-01	3.22E-01	**4.20E-02**	8.34E-01	3.48E-01
	5	Best	3.59E-02	1.27E-03	1.98E-02	2.09E-02	3.05E-02	**1.13E-03**	3.15E-02	1.07E-01	1.33E-01	3.05E-02
		Mean	5.19E-01	**2.52E-03**	1.72E-01	4.18E-02	4.09E-02	5.79E-02	1.08E-01	4.91E-01	3.18E-01	4.81E-02
		Worst	1.70E-01	**5.60E-03**	9.57E-01	6.60E-01	4.89E-02	1.51E-01	3.28E-01	1.08E+00	1.53E+00	2.24E-01
	8	Best	3.05E-02	3.51E-01	4.40E-02	5.27E-02	**2.21E-02**	1.30E-01	2.39E-02	2.61E-02	2.36E-02	1.71E-01
		Mean	6.57E-01	6.99E-01	4.77E-01	7.88E-01	4.87E-02	6.38E-01	1.32E-01	**4.73E-02**	1.61E-01	3.42E-01
		Worst	2.87E+00	8.35E-01	2.79E-01	1.01E+00	3.89E-01	1.56E+00	2.36E-01	**1.21E-01**	2.89E-01	4.86E+00
	10	Best	**1.09E-02**	3.47E-01	6.38E-01	1.16E-02	3.22E-02	2.29E-01	2.71E-02	1.01E-01	1.88E-01	2.00E-01
		Mean	6.21E-01	6.86E-01	7.10E-01	3.59E-02	3.12E-01	2.36E+00	**3.41E-02**	7.56E-01	6.02E-01	7.47E-01
		Worst	7.10E-01	8.53E-01	9.14E+00	**1.08E-01**	4.10E-01	3.02E+00	2.42E-01	1.09E+00	3.02E+00	9.48E-01

The values in boldface indicate better results.

Biographies

Sandeep U. Mane is an Assistant Professor at the Department of Computer Science and Engineering, Rajarambapu Institute of Technology, Rajaramnagar, MS, India. He is a PhD Research Scholar at Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation (Deemed to be University), Vaddeswaram, Guntur Dist., AP, India. He has received BSc Engineering in Information Technology from BVCOE, Kolhapur, MS, India and Master of Technology (MTech) in Computer Engineering from Dr. BATU, Lonere, MS, India. His research interests include multi-objective and many-objective optimization problems, evolutionary algorithms, and high-performance computing using GPGPU.

Manda Rama Narasinga Rao is a Professor at the Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation (Deemed to be University), Vaddeswaram, Guntur Dist., AP, India. He received a Doctor of Philosophy in Computer Science and Systems Engineering from Andhra University, Visakhapatnam, Andhra Pradesh, India in 2012. His research interests include Applications of Neural Networks, Content-Based Information Retrieval, Automata Theory, Bio-informatics, and Software Engineering.