International Journal of Engineering Trends and Technology
ISSN: 2231–5381 / https://doi.org/10.14445/22315381/IJETT-V72I7P118

Volume 72 Issue 7, 168-177, July 2024

Original Article

A Ranking Policy Based on Many Objective Optimization

Pratyusha Rakshit[1], Archana Chowdhury[2]

[1]*Electronics & Telecommunication Department, Jadavpur University, Kolkata.*
[2]*Computer Science & Engineering Department, Christian College of Engineering & Technology, Bhilai.*

[2]*Corresponding Author : chowdhuryarchana@gmail.com*

Received: 16 July 2023 Revised: 03 October 2023 Accepted: 18 June 2024 Published: 26 July 2024

Abstract - *Many objective optimizations have more competing objectives than multi-objective optimizations (MOO); thus, they are more challenging to solve**.** In order to solve the many objective optimizations (MaOO), a novel strategy based on a ranking policy is put forth in this study. In many objective optimizations, a solution might not be effective for all goals; hence a new ranking scheme is suggested in place of pareto ranking. Artificial Bee Colony (ABC) is the algorithm that was employed in this study. The procedure is initially conducted in parallel with each of the multiple objective optimization problem's objectives. The following phase involves sifting through and choosing the high-quality solutions that are produced by simultaneously optimizing each of the multiple objectives. The proposed ranking system is used to grade the constituents of high-quality solutions. Performance was assessed using DTLZ and WFG tests, and the findings imply that the suggested approach performs better than cutting-edge techniques.*

Keywords - *Various objective optimization, Multi-objective optimization, Fitness function, Artificial Bee Colony, Pareto ranking.*

1. Introduction

Many-objective optimization complications have been a topic of research in recent years. A subset of Multi-Objective Optimization (MOO) is known as a Many-Objective Optimization (MaOO) problem. The term "Multi-Objective Optimization Problems" (MOOPs) refers to problems with three or fewer objectives, whereas "many-objective optimization problems" (MaOOPs) refer to problems with more than three objectives [1]. Researchers are taking much interest in developing robust MaOO algorithms because of their possible application in varied areas[2-4]. V Pareto introduced the idea of multi-objective optimization in 1896 by generalizing a number of difficult objectives into an optimization problem. Schaffer introduced the many objective optimization evolutionary algorithm MaOEA in 1975 [5]. The traditional evolutionary MOO methods are useful for efficiently locating the global optima of a multiobjective optimization problem with two or three approaches. But once the quantity of objectives rises, as it does in a MaOO situation, nearly every member of the population becomes non-dominated. Therefore, multi-objective evolutionary algorithms designed to address multi-objective optimization problems were applied to many issues related to optimization schemes. The number of non-dominated solutions grows exponentially when several objective optimization issues are attempted to be resolved, and multi-objective algorithms based on dominance relationships become less capable of searching. Because of this, the conventional EMOO algorithms, which select solutions based on Pareto ranking non-supremacy criteria [6], are unable to guarantee that the algorithm will converge to the ideal Pareto front.

Only a subset of objectives are included in each generation of a candidate solution and the performance evaluation of the solution set, according to the dimensionality reduction base algorithm, which has been the recent trend to overcome the challenges faced by traditional evolutionary algorithms in solving many-objective optimization problems [7-9].In this case, the subset of objectives is constructed so that only the objectives that conflict are recognized. A portion of MaOO's research focuses on algorithms based on dominance relations, in which selection is based on the dominance rule. The diversity maintenance technique and improved Pareto dominance connection guarantee the algorithm's convergence and diversity. ε-dominance [10], fuzzy Pareto dominance [11], and subspace dominance comparison [12] are a few dominance relation-based approaches. In addition to the solutions based on dominance relations and dimensionality reduction, there is evidence of alternative approaches to address MaOO problems [13], [14]. The population-based evolutionary algorithm's latent parallelism, which may have hastened the algorithm's convergence, has not been taken into account by the strategies now in use.

This paper grants a novel scheme for parallel optimization of the L objectives in the direction of solving a MaOO problem. It is further practical to advance a MaOO problem's solution with respect to a single objective, as a single solution might not be able to achieve all L objectives optimally. We have, therefore, used N evolutionary algorithms to improve the individual N objectives in parallel, effectively utilizing the built-in parallelism of population-based search. The fittest trial solutions objective function value is noted to be f_k^{best} as soon as the N optimization algorithms have reached convergence. A set μ_k is created by keeping the top $100\times(1-\beta)\%$ of the population's best-fit solutions for each objective for which $0 < \beta < 1$. μ_k is the set made up of the best solution and a few solutions that, at most, are $100\times\beta\%$ less than the best solution. Next, we take the union of all the sets μ_ks. At least one objective function is satisfied by every solution in the union set to a degree of $100\times(1-\beta)\%$ or greater.

With a threshold-based ranking policy, the MaOO problem's global optima—which maximizes the optimization of all L objectives—is found. The union set's solutions are rated according to the adopted policy. The policy's initial goal is to carefully find all L objectives' solutions within the union set whose objective function values fall within the interval $[f_k^{best}, (1+\beta) f_k^{best}]$. Rank 1 is given to all these solutions. All of these solutions will add up to a total rank of N. Based on all relevant objectives equally, each of these solutions represents the MaOO problem's comprehensive ideal situation. Alternatively, in the event that no such solution is discovered, the policy looks for solutions with fitness in the range $[f_k^{best}, (1+\alpha) f_k^{best}]$ for (L−1) objectives with Rank 1 and solutions with fitness in the range $[(1+\beta) f_k^{best}, (1+2\beta) f_k^{best}]$ for the remaining objectives, which are given Rank 2 The resulting solutions, which have a rank sum of N+1, are regarded as the MaOO problem's second-finest universal optimal. Nonetheless, in the event that the policy is unable to locate these solutions once more, a different set of solutions is located by commissioning explanations having the range of fitness value as $[f_k^{best}, (1+\beta) f_k^{best}]$for (L−2) objectives, with Rank 1, and solutions with fitness values in the range $[(1+\beta) f_k^{best}, (1+2\beta) f_k^{best}]$ for the remaining two objectives with Rank 2. The procedure might go on until workable answers to the MaOO problem are not found. Our intention is to select solutions (from the union set) that have the lowest rank sum. Remarkably, equal rank sum solutions for the MaOO problem are considered to have comparable global optima.

The Artificial Bee Colony algorithm is used to understand the performance of the proposed MaOO algorithm. ABC].Many-objective optimization using Artificial Bee Colony with Temporal Deference Q learning is the name given to the suggested MaOO algorithm. MaOABC-TDQL. Three cutting-edge methods [16-18] are compared with the proposed algorithm in the direction of optimizing an eminent seven DTLZ [19] in addition to nine WFG [20] various-objective standard difficulties. We take into consideration 6, 8, and 10 objectives for each benchmark problem. It is intended for widely held standard purposes, the suggested procedure accomplishes in an improved way than alternative procedures in terms of inverted generational distance [18], [22], and hypervolume [24]. These differences are statistically significant. The research is alienated into five segments. Segment II overviews ABC procedure. Segment III offers the MaOO technique using the ABC algorithm. Investigational situations for the standards and simulation approaches are described in Segment IV. Segment V concludes the paper.

2. Artificial Bee Colony (ABC) Optimization Algorithm

In the ABC procedure, the cluster of artificial bees comprises three clusters of bees:

- In the dancing area, onlooker bees are waiting to select a food basis.
- Employed bees: they go to the food bases that they have already visited.
- Scout bees: they randomly explore for food basis.

The location of a food basis characterizes a potential solution to the optimization problem in the ABC procedure, and the quantity of nectar a food basis produces signifies the fitness of the associated solution. The amount of solutions in the population is equal to the amount of employed bees and onlooker bees. ABC consists of the subsequent actions.

2.1. Initialization

ABC produces an arbitrarily dispersed preliminary population P (G=0) of N_p solutions (food source positions). The size of the population is denoted by N_p. Each solution X_i (i=0, 1, 2,…, N_p -1) is a D dimensional vector.

2.2. Employed Bees Placement

Using equation (2), an employed bee regulates the location in her retention based on local statistics (visual information) and verifies the quantity of nectar from the novel foundation. The bee forgets the old location and learns the new one if the amount of nectar from the new source is greater than that from the old one. In any other case, the bee remembers where the previous one was.

2.3. Onlooker Bees Placement

An onlooker bee selects a food source after taking into account the nectar information of every employed bee in operation. The probability value, p_i, connected to that food source determines which food source should be chosen. The formula to get p_i is as follows:

$$P_i = \frac{f_i}{\sum_{j=0}^{j=Np-1} f_j} \quad (1)$$

Where f_i is the value of fitness of i^{th} solution assessed by its employed bee. Subsequently, the onlooker bee modifies its memory to match the employed bee's position and retains a

superior food source location. A solution $X_i^{/}$ in the neighborhood of X_i is found. Parameters j and X_m are chosen arbitrarily for solution $X_i^{/}$. In the solution $X_i^{/}$, all the parameter values of solution X_i are copied except for parameter j; for example,

$$X_i^{/} = (x_{i0}, x_{i1}, \ldots, x_{i(j-1)}, x_{ij}^{/}, x_{i(j+1)}, \ldots, x_{i(D-1)}).$$

The parameter value of $x_{ij}^{/}$ in solution $X_i^{/}$ is calculated accordingly:

$$X_{ij}^{/} = x_{ij} + u(x_{ij} - x_{mj}) \quad (2)$$

Where u is a uniform variable in [-1, 1] and m is any quantity between 0 to N_p-1 but not equivalent to i.

2.4. Placement of Scout Bee

If, after a particular quantity of cycles recognized as the "limit", no improvement is observed in the position, the food source is dropped from consideration in the ABC algorithm. The scouts replace this abandoned food source by drawing a position at random. After that again, steps (2), (3), and (4) are recurring until the stopping criteria are met.

3. Artificial Bee Colony Induced Many-Objective Optimization

A two-step solution is provided to the MaOO problem having L objectives, through this research paper. First, L ABCs optimize each of the L objectives independently and concurrently. Finding the solutions that best meet the entire L objectives, is the focus of the second step. Below is a description of the two steps.

3.1. Individual Parallel Optimization of L Objectives

Here, the ABC algorithm is used to optimize each of the L individual objectives in parallel. Individual optimisation is based on the fundamental idea that it is improbable that a potential solution will be performing competently in terms of all objectives. Therefore, it makes sense to evolve a person only in relation to a chosen goal for which it has the best chance of success.

3.1.1. Initialization

An arbitrarily distributed preliminary population P^k (G=0) of N_p solutions (food source positions) is generated for the k^{th} objective. N_p here signifies the population size. Each solution of the form X_i^k (i=0, 1, 2,…, N_p -1) is a D-dimensional vector.

3.1.2. Placement of Employed Bees

An employed bee tests the amount of nectar from a new source and adjusts the position based on local data, as stated in equation (4) for the k^{th} objective. Only when the nectar value is greater than the earlier one does the employed bee memorize the new position; otherwise, it retains the prior position in its memory.

3.1.3. Placement of Onlooker Bees

After analysing the nectar data from every employed bee, an onlooker bee selects the food source for the k^{th} objective on the basis of probability p_i^k. The value of p_i^k is given as

$$P_i^k = \frac{f_i^k}{\sum_{j=0}^{j=Np-1} f_i^k} \quad (3)$$

Where f_i^k is the fitness value of the solution i, for k^{th} objective as evaluated by its employed bee. The onlooker bee then makes a change in location and commits the location of the superior food source to memory. To find a solution $X_i^{/k}$ in the neighbourhood of X_i^k, parameters j and X_m are randomly selected. In the solution $X_i^{/k,}$ except for the selected parameter j, all other parameter values are the same as in the solution X_i^k; for example,

$$X_i^{/k} = (x_{i0}^k, x_{i1}^k, \ldots, x_{i(j-1)}^k, x_{ij}^{/k}, x_{i(j+1)}^k, \ldots, x_{i(D-1)}^k).$$

Equation (4) represents the way to calculate the value of the $x_{ij}^{/k}$ parameter in the $X_i^{/k}$ solution:

$$X_{ij}^{/k} = x_{ij}^k + u(x_{ij}^k - x_{mj}^k) \quad (4)$$

Where u is a uniform variable in [-1, 1] and m is any number among 0 to N_p-1 but not equal to i.

3.1.4. Placement of Scout Bee

When a situation can no longer be enriched after certain cycles, known as the "limit," the food source is given up, and the scouts substitute it by selecting a position at random. Till the stopping criteria are met, steps (2), (3) and (4) will be repeated. In parallel, we use L such ABCs, each of which handles the optimization of the k^{th} objective, where k = [1, L].

3.2. Union of Candidate Solutions

The next step is to find the set of common solutions for the L optimizing objectives, as determined by the convergence of all k-ABCs for k = [1, L]. After k-ABC has stabilized and is optimizing the particular objective function fk(.) for k = [1, L], the fittest trial solution's best (or least) objective function value is indicated by the symbol f_k^{best}.It seems that in the MaOO scenario, where k, r ∈[1, L], but k ≠ r, $\vec{X}^{k-best}$ may not always be the best solution in accordance with the r-th objective. Conversely, a solution $X_i^k \in P^k$ that is marginally less than $\vec{X}^{k-best}$ could potentially solve all of the outstanding (L–1) optimization problems. This led us to think about a collection of solutions μ_k rather than a single ideal solution, having objective function value in the range [f_k^{best}, (1+ β) f_k^{best}]. Here, the constant 0 < β <1 is user-defined. To begin with, μ_k is constituted with the top 100×(1− β)% best-fit solutions for the k-th objectives. The parameter is to be set as β = 0.05 if we wish to incorporate the top 95% fittest solutions with respect to the k-th objective into μ_k. It appears that the members in μ_k will have fk(.) measures in the range [f_k^{best}, 1.05 f_k^{best}]. The best-fit solution $\vec{X}^{k-best}$, will be permitted to go in the μ_k if β = 0. The μ_k would be more liberal the higher the value of β.

Symbolically,

$$\mu_k = \left\{ \vec{X} | f_k(\vec{X}) \epsilon \left[f_k^{best}, (1 + \beta) f_k^{best} \right] and\ \vec{X} \epsilon \bigcup_{g=1}^{G\_max} P^k(g) \right\} \quad (5)$$

This is followed for every concerned objective (k = [1, L]) in the MaOO problem. A combination of the topmost 100×(1− β)% finest solutions of the distinct L objective function is then taken. Let φ represent the union set. It is given as follows:

$$\varphi = \mu_1 \cup \mu_2 \cup \ldots\ldots \cup \mu_L \quad (6)$$

To effectively find the trial solution that will outstandingly optimize every L objective of the MaOO problem, a ranking policy is adopted.

3.2.1. Step 1: Ranking the Members of φ

This stage involves ranking each member of the union set φ, according to the unique values for the L objective function. Every L objective is assessed for every member of φ. Regarding the k-th objective, the solutions of φ having the objective function in the range [f_k^{best}, (1+ β) f_k^{best}] are ranked 1. It states that, with regard to k-th-objective function measure fk(.), the first ranked solutions of φ optimize fk(.) at most by 100% and at least by 100×(1− β)%. With respect to the k-th objective, the solutions of φ are ranked 2 if the objective values are within [(1+ β) f_k^{best}, (1+2 β) f_k^{best})]. For instance, if β = 0.05, then fk(.) is optimised by the second-category members of φ (relative to fk(.) only) by atleast 100×(1−2 β) = 100×(1–2×0.05) % = 90% and, by at most, 100×(1− β) = 100×(1–0.05) % = 95%. Correspondingly, the solutions of Ω with fitness inside [(1+2 β) f_k^{best}, (1+3 β) f_k^{best}) are ranked 3, taking into consideration only the k-th objective. This is repetitive up until whole constituents of the combination set φ are ranked as 1, 2, 3, and so on, in consideration of the fk(.) objective alone.

Briefly, a solution $\vec{X} \in \varphi$ is allocated a rank $r_k(\vec{X}) = R$ in accordance to f_k(.) if

$$f_k(\overrightarrow{X}) \in \left[\left((1 + (R-1)\beta) f_k^{best}, (1 + R\beta) f_k^{best} \right) \right] \quad (7)$$

The above-mentioned technique is then repeated for all objectives k = [1, L]. The sum of rank(SR) is then obtained subsequently after procuring the rank. $r_k(\vec{X})$ of $\vec{X} \in \varphi$ for all objectives k = [1, L]. The value of SR is given as,

$$SR(\vec{X}) = \sum_{k=1}^{L} r_k(\vec{X}) \quad (8)$$

3.2.2. Step 2: Creating Subsets of φ with Members having Equal SR

In this step, the ideal set of equally good solutions that maximizes the optimization of all L objectives is determined. Evidently, the (approximate) global optimal of the MaOO problem is represented by the solutions of φ, which attain a rank 1 with respect to all objectives fk(.) for k = [1, L]. All such solutions are placed in the subset φ_1. The sum of the rank of the candidates of φ_1 is L, as all of them possess rank 1 for all individual objectives. Subset φ_2 is identified if φ_1 is empty, i.e., if no solution has fitness within 100(1− β)% of the highest fitness value for all distinct L objectives. Subset φ_2 consists of solutions of φ that have rank 1 for any of L−1 objectives (amongst f1(.) to fL(.)) and rank 2 for the remaining one objective. Therefore, the SR of the members of φ_2 will be (L−1)×1+2 = (L+1).

However, if both φ_1 and φ_2 are vacant, another set is created with solutions having rank 1 in accordance with L−2 objectives and rank 2 in accordance with two objectives. The SR of these solutions will be (L−2)×1+2×2 = (L+2). Unfortunately the solutions with high value of SRs are reasonably inferior. This procedure undergoes until the creation of a non-empty set of solutions representing the possible global optima of the objectives under consideration. The set φ^A of food sources with minimum SR represents the approximate global optima of the problem.

4. Experimental Results

4.1. Benchmark Functions

The suggested MaOABC algorithm's performance is examined with respect to DTLZ [19] and WFG [20]. DTLZ and WFG are the benchmark suites. The DTLZ test suite contains seven benchmark functions, from DTLZ1 to DTLZ7. The efficiency of the suggested MaOABC is tested by keeping different numbers of objectives L, such as L=6,8, and 10. The number of variables D is set to L + k – 1. Parameter k is set to 5 for DTLZ1 and 20 for DTLZ7, while it is set to 10 for the residual 5 benchmark problems in the DTLZ being tested.WFG1 through WFG9, the nine benchmark functions, are also utilized in the analysis of the performance of the MaOABC algorithm. Here, K + M gives the number of decision variables (D). In accordance with the number of objectives L = 6, 8, and 10, the distance parameter M is set constant at 10, and the position parameter K has corresponding values of 10, 7, and 9 [21].

4.2. Comparative Framework and Parameter Setting

Three well-known MaOO algorithms—the grid-based evolutionary algorithm (GrEA) [18], the multi-objective evolutionary algorithm based on decomposition (MOEA/D) [17], and the hypervolume estimation algorithm for multiobjective optimization (HypE) [16]—are compared to the suggested MaOABC algorithm. Each of these three competitor algorithms has a distribution index of 20. These algorithms use a polynomial mutation strategy with probability 1/D and simulated binary crossover with possibility 1. For scalarizing, the Tchebycheff function is used in MOEA/D, where the neighbourhood size is 10% of the entire population.

Table 1. Population size of MOEA/D

N	(p1, p2)	Population Size
6	(4, (4,1)	132
8	(3, 2)	156
10	(3,2)	275

The number of reference points inside the Pareto front and along the boundary, which are determined by two parameters, p1 and p2, respectively, as shown in Table I, determines the size of the MOEA/D population. The population sizes of HypE, GrEA, and MaOABC are likewise set. β is a significant parameter that governs MaOABC's performance. Section IIIB makes clear that significantly worse members can have a substantial impact on the MaODE's performance at higher values of β (near unity).

Therefore, having a small value of β is desirable. Nevertheless, in the event that β = 0, the set of optimal solutions will become less diverse as here, only the L best individuals recognised by running L optimization algorithms in parallel will be utilized. β is progressively decreased from 1 to 0 at a decreasing period of 0.005 in order to find the optimal setting. Experimental observations show that performance does not significantly change for $\beta \leq 0.05$. Thus, we have set $\beta = 0.05$.

4.3. Performance Metrics

The performance metrics used to authenticate the efficacy of the MaOABC algorithm are as follows:

4.3.1. Inverted Generational Distance

Let ξ^* be a collection of consistently dispersed points in the objective space that are situated along the MaOO problem's optimal Pareto front. Let ξ^{app} be a rough estimate of the optimal Pareto front that captures the fitness values of the solution having the lowest SR in the MaOABC perspective. The inverted generational remoteness (IGD) amongst ξ^* and ξapp [22] is as follows:

$$IGD(\xi^{app}, \xi^*) = \frac{\sum_{u \in \xi^*} d(u, \xi^{app})}{|\xi^*|} \quad (9)$$

Now $d(u, \xi^{app})$ indicates the least Euclidean distance amid the points $u \in \xi^*$ and the points in ξ^{app}. A low value of IGD confirms that ξ^{app}, attained by the anticipated MaOO procedure, is very adjacent to the best Pareto front ξ^*. Here, the number of points is fixed at 500, consistently sampled across the true Pareto front.

4.3.2. Hypervolume

The entire area of the objective space, which is dominated by the members of $\xi^{app,}$ is represented by the hypervolume (HV) of ξ^{app}. In the objective space, the hypervolume HV(ξ^{app}) is assessed relative to a user-provided reference point. The reference point denotes the worst-case point or the anti-optimal points [15] in the objective space.

Using the Monte-Carlo method from [24], the HV(ξ^{app})is obtained. In the objective space, a hyper-rectangle is taken into consideration amongst pre-defined reference points and the origin in the objective space. The reference point chosen in this instance is (1, 1,..., 1) [21]. In order to preserve the unvarying measure, the objective function value of the WFG test suite is normalized in the interval [0,1] prior to estimating the hypervolume. In this hyper-rectangle, 106 sampling points are taken into account through Monte-Carlo simulation. The section of sample points dominated by the ξ^{app} within the hyper-rectangle is given by HV(ξ^{app}). The attainment function for all sampled points u in the hyper-rectangle is given below,

$$\alpha(u) = \begin{cases} 1, & if\ \xi^{app}\ dominates\ u \\ 0, & otherwise \end{cases} \quad (10)$$

The average values of the attainment function are calculated over the entire sample points in the hyper-rectangle to determine the hypervolume indicator HV(ξ^{app}) [1]. It seems that the MaOO algorithm performs better for higher HV values. For a MaOO problem, a front ξ_1^{app} dominates another front ξ_2^{app} if HV(ξ_1^{app}) > HV(ξ_2^{app}

4.4. Comparative Analysis

The values in Table II represent the interquartile range (IQR) and median of the IGD metric. These values are obtained by 50 separate runs of each benchmark function of the test suite DTLZ.

Similarly, Table III displays the IQR and median of the HV metric resulting from 50 separate runs. The IQR is shown in parenthesis beneath the median value in Tables II and III. Bold text indicates the best value of the metric.

Table 2. Comparison analysis of performance with respect to IGD

Functions	*N*	HypE	MOEA/D	GrEA	MaOABC
DTLZ1	6	2.292e-01	1.085e-01	1.373e-01	**9.926e-02**
		(4.357e-02)	(5.856e-02)	(6.745e-03)	**(3.866e-03)**
		[1.173e-02]	[2.876e-02]	[1.765e-02]	**[NA]**
	8	3.148e-01	1.866e-01	2.511e-01	**1.587e-01**
		(2.534e-02)	(1.377e-02)	(3.665e-02)	**(1.256e-02)**
		[2.151e-03]	[8.450e-03]	[4.106e-02]	**[NA]**
	10	2.378e-01	2.218e-01	**1.395e-01**	1.537e-01
		(3.667e-02)	(3.333e-02)	(**1.786e-02**)	(2.963e-02)

		[3.238e-02]	[3.245e-02]	**[NA]**	[5.749e-02]
DTLZ2	6	4.697e-01	4.066e-01	2.574e-01	**2.748e-01**
		(5.360e-02)	(4.701e-02)	(1.894e-03)	**(1.825e-03)**
		[1.481e-02]	[2.735e-02]	[5.023e-02]	**[NA]**
	8	7.017e-01	6.644e-01	3.669e-01	**3.715e-01**
		(4.494e-02)	(5.700e-02)	(3.383e-03)	**(3.692e-03)**
		[1.842e-02]	[3.128e-02]	[5.033e-02]	**[NA]**
	10	7.329e-01	7.497e-01	3.950e-01	**1.689e-01**
		(4.447e-02)	(6.853e-02)	(2.600e-02)	**(2.362e-02)**
		[3.901e-02]	[3.878e-02]	[4.646e-02]	**[NA]**
DTLZ3	6	4.487e-01	4.937e-01	**2.167e-01**	**2.361e-01**
		(7.654e-02)	(1.910e-01)	**(6.232e-02)**	**(7.483e-02)**
		[2.079e-02]	[2.233e-02]	**[NA]**	**[NA]**
	8	6.344e-01	7.754e-01	**3.778e-01**	4.374e-01
		(1.227e-01)	(1.287e-01)	**(1.982e-01)**	(4.385e-02)
		[2.553e-02]	[3.974e-02]	**[NA]**	[4.219e-02]
	10	7.369e-01	6.874e-01	4.959e-01	**3.984e-01**
		(5.950e-02)	(2.717e-01)	(1.517e-01)	**(7.692e-02)**
		[2.664e-02]	[3.221e-02]	[4.057e-02]	**[NA]**
DTLZ4	6	6.383e-01	5.268e-01	**2.658e-01**	2.782e-01
		(1.209e-01)	(1.679e-02)	**(2.210e-03)**	(1.537e-03)
		[2.750e-02]	[4.379e-02]	**[NA]**	[5.695e-02]
	8	7.033e-01	7.596e-01	3.783e-01	**3.614e-01**
		(7.503e-02)	(6.516e-02)	(2.881e-03)	**(2.863e-03)**
		[3.112e-02]	[1.506e-02]	[4.935e-02]	**[NA]**
	10	8.256e-01	7.810e-01	4.040e-01	**3.754e-01**
		(1.769e-02)	(2.023e-02)	(9.373e-03)	**(1.483e-02)**
		[1.152e-02]	[2.354e-02]	[4.221e-02]	**[NA]**
DTLZ5	6	**1.530e-02**	2.412e-02	8.920e-02	7.517e-02
		(9.717e-03)	(1.539e-02)	(1.586e-02)	(1.587e-02)
		[NA]	[3.138e-02]	[1.129e-02]	[4.262e-02]
	8	5.020e-02	8.441e-02	1.703e-01	**3.374e-02**
		(1.022e-02)	(1.922e-02)	(4.758e-02)	**(3.478e-02)**
		[2.151e-02]	[1.555e-02]	[4.616e-02]	**[NA]**
	10	**2.934e-02**	7.163e-02	4.155e-01	2.811e-01
		(**2.917e-03**)	(4.168e-02)	(7.877e-02)	(5.873e-02)
		[NA]	[4.898e-02]	[2.194e-02]	[6.198e-02]
DTLZ6	6	3.557e-01	**1.587e-01**	2.520e-01	2.201e-01
		(5.688e-02)	**(3.154e-01)**	(6.178e-02)	(1.623e-01)
		[1.311e-02]	**[NA]**	[2.043e-02]	[2.986e-02]
	8	4.522e-01	**1.658e-01**	5.590e-01	2.372e-01
		(3.403e-01)	**(2.918e-02)**	(1.735e-01)	(4.261e-02)
		[3.014e-02]	**[NA]**	[1.108e-02]	[3.655e-02]
	10	2.320e-01	4.624e-01	4.116e-01	**2.382e-01**
		(4.720e-02)	(5.534e-02)	(2.717e-01)	**(4.167e-02)**
		[5.593e-02]	[2.539e-02]	[2.120e-02]	**[NA]**
DTLZ7	6	1.555e+00	**4.024e-01**	5.602e-01	5.867e-01
		(1.276e-01)	**(1.652e-02)**	(2.501e-02)	(4.698e-03)
		[4.276e-03]	**[NA]**	[4.005e-02]	[1.422e-02]
	8	2.831e+00	1.342e+00	1.001e+00	**9.422e-01**
		(2.222e-01)	(2.423e-01)	(2.334e-02)	**(1.964e-02)**

		[2.892e-02]	[3.651e-02]	[4.644e-02]	**[NA]**
	10	2.960e+00	1.543e+00	1.321e+00	**1.362e+00**
		(1.684e-01)	(4.554e-01)	(9.439e-02)	(**8.910e-02**)
		[2.294e-02]	[2.734e-02]	[4.815e-02]	**[NA]**

Table 3. Comparative analysis of performance with respect to HV for WFG1 to WFG9

Functions	***N***	**HypE**	**MOEA/D**	**GrEA**	**MaOABC**
WFG1	6	9.218e-01	9.677e-01	9.304e-01	**9.901e-01**
		(9.004e-03)	(2.602e-02)	(3.901e-02)	**(2.582e-02)**
		[1.345e-02]	[5.827e-02]	[4.587e-02]	**[NA]**
	8	9.161e-01	9.045e-01	9.040e-01	**9.263e-01**
		(4.489e-02)	(4.749e-02)	(5.703e-02)	**(4.154e-02)**
		[5.433e-02]	[2.437e-02]	[1.881e-02]	**[NA]**
	10	9.108e-01	**9.488e-01**	8.493e-01	8.524e-01
		(1.148e-01)	**(1.568e-02)**	(3.669e-02)	(5.818e-02)
		[2.733e-02]	**[NA]**	[3.222e-02]	[3.526e-02]
WFG2	6	6.676e-01	8.495e-01	9.348e-01	**9.719e-01**
		(9.680e-02)	(8.527e-02)	(7.229e-02)	(**2.634e-02**)
		[3.238e-02]	[3.395e-02]	[4.725e-02]	**[NA]**
	8	**9.920e-01**	9.710e-01	9.414e-01	9.892e-01
		(**2.610e-03**)	(2.178e-02)	(9.894e-02)	(3.121e-03)
		[NA]	[1.044e-02]	[1.181e-02]	[5.586e-02]
	10	9.878e-01	9.734e-01	9.572e-01	**9.962e-01**
		(2.983e-03)	(3.971e-02)	(1.780e-02)	(**1.958e-03**)
		[5.309e-02]	[5.036e-02]	[2.293e-02]	**[NA]**
WFG3	6	5.420e-01	4.827e-01	**5.881e-01**	5.651e-01
		(7.047e-03)	(6.446e-03)	**(8.292e-03)**	(7.421e-03)
		[2.080e-02]	[3.310e-02]	**[NA]**	[5.862e-02]
	8	5.933e-01	**5.938e-01**	5.730e-01	5.923e-01
		(1.739e-02)	(**5.275e-03**)	(1.316e-02)	(2.425e-02)
		[6.067e-02]	**[NA]**	[4.164e-02]	[5.289e-02]
	10	5.538e-01	**6.018e-01**	4.966e-01	5.818e-01
		(4.830e-03)	**(2.150e-03)**	(1.616e-02)	(3.645e-03)
		[2.703e-02]	**[NA]**	[4.349e-02]	[5.891e-02]
WFG4	6	7.495e-01	7.516e-01	**8.026e-01**	7.887e-01
		(2.298e-02)	(1.347e-02)	(**1.071e-02**)	(1.672e-02)
		[1.590e-02]	[3.227e-02]	**[NA]**	[4.729e-02]
	8	7.296e-01	6.535e-01	7.879e-01	**8.511e-01**
		(3.030e-02)	(1.976e-02)	(8.003e-03)	(**6.714e-03**)
		[2.723e-02]	[3.196e-02]	[3.236e-02]	**[NA]**
	10	5.988e-01	4.787e-01	**8.114e-01**	7.842e-01
		(5.458e-02)	(1.629e-02)	(**8.871e-03**)	(1.216e-02)
		[2.719e-02]	[3.605e-02]	**[NA]**	[6.283e-02]
WFG5	6	6.498e-01	5.315e-01	**7.035e-01**	7.236e-01
		(1.668e-02)	(1.187e-02)	(**5.934e-03**)	(7.482e-03)
		[5.290e-03]	[5.485e-03]	**[NA]**	[1.259e-02]
	8	5.657e-01	4.125e-01	6.572e-01	**7.264e-01**
		(1.317e-02)	(1.185e-02)	(5.914e-03)	**(6.287e-03)**
		[2.022e-02]	[2.241e-02]	[3.817e-02]	**[NA]**
	10	3.760e-01	3.333e-01	5.843e-01	**6.214e-01**
		(1.535e-02)	(8.460e-03)	(5.784e-03)	(**4.819e-03**)
		[3.859e-02]	[4.664e-02]	[4.863e-02]	**[NA]**
WFG6	6	4.585e-01	6.557e-01	**6.945e-01**	5.926e-01
		(7.447e-03)	(3.900e-02)	**(2.380e-02)**	(2.158e-02)

		[6.944e-03]	[9.601e-03]	**[NA]**	[5.511e-02]
	8	2.620e-01	4.920e-01	6.157e-01	**6.828e-01**
		(1.833e-02)	(2.282e-02)	(1.590e-02)	**(1.513e-02)**
		[2.627e-02]	[2.651e-02]	[4.305e-02]	**[NA]**
	10	3.969e-01	4.593e-01	7.704e-01	**7.161e-01**
		(1.068e-02)	(3.319e-02)	(1.930e-02)	**(1.218e-02)**
		[2.600e-02]	[3.357e-02]	[5.706e-02]	**[NA]**
WFG7	6	3.656e-01	5.924e-01	6.521e-01	**6.276e-01**
		(1.657e-02)	(3.111e-02)	(2.662e-02)	(**7.892e-03**)
		[1.310e-02]	[1.738e-02]	[2.930e-02]	**[NA]**
	8	6.242e-01	5.496e-01	7.367e-01	**7.876e-01**
		(4.410e-02)	(3.474e-02)	(8.555e-03)	**(8.325e-03)**
		[1.213e-02]	[2.212e-02]	[3.774e-02]	**[NA]**
	10	6.300e-01	3.069e-01	8.625e-01	**8.792e-01**
		(3.549e-02)	(8.094e-03)	(7.448e-03)	**(6.562e-03)**
		[1.973e-02]	[3.439e-02]	[3.681e-02]	**[NA]**
WFG8	6	3.776e-01	2.065e-01	**4.481e-01**	4.427e-01
		(1.519e-02)	(1.978e-02)	**(3.052e-02)**	(2.621e-02)
		[3.417e-02]	[2.211e-02]	**[NA]**	[3.680e-02]
	8	2.746e-01	3.129e-01	4.322e-01	**6.231e-01**
		(1.708e-02)	(2.249e-02)	(1.650e-02)	(**1.624e-02**)
		[1.351e-02]	[1.654e-02]	[2.121e-02]	**[NA]**
	10	3.001e-01	5.299e-01	**7.937e-01**	5.821e-01
		(2.038e-02)	(2.620e-02)	**(3.604e-02)**	(1.623e-02)
		[2.149e-02]	[4.108e-02]	**[NA]**	[4.538e-02]
WFG9	6	3.436e-01	3.935e-01	**7.097e-01**	6.932e-01
		(3.511e-02)	(4.076e-02)	**(4.352e-02)**	(4.530e-02)
		[3.775e-02]	[3.845e-02]	**[NA]**	[4.132e-02]
	8	4.126e-01	4.880e-01	5.650e-01	**6.395e-01**
		(6.416e-02)	(2.691e-02)	(1.504e-02)	**(1.198e-02)**
		[1.887e-02]	[3.952e-02]	[4.746e-02]	**[NA]**
	10	4.348e-01	5.563e-01	**7.640e-01**	6.910e-01
		(3.707e-02)	(3.626e-02)	**(6.116e-03)**	(5.911e-03)
		[2.193e-02]	[3.356e-02]	**[NA]**	[4.243e-02]

Table 4. Friedman ranks with respect to IGD metric for the DTLZ test suite

Algorithms	**Friedman Rank**
MaOABC	**1.7000**
GrEA	2.3571
MOEA/D	2.8571
HypE	3.2857
Friedman Statistics (3 DOF)	22.2429 (Critical Value: 7.815)
Iman-Davenport Statistics (3, 60 DOF)	10.9148 (Critical Value: 3.340)

Table 5. Friedman ranks with respect to HV Metric for WFG test suite

Algorithms	**Friedman Rank**
MaOABC	**1.8976**
GrEA	2.1111
MOEA/D	3.0370
HypE	3.2222
Friedman Statistics (3 DOF)	27.8444 (Critical Value: 7.815)
Iman-Davenport Statistics (3, 78 DOF)	13.6196itical Value: 3.280)

The statistical significance level of the IGD metric of the best algorithm and any one of the remaining three competitive algorithms for 50 sample values is displayed in the third bracket below the IQR value in Table II. The value is found using the Wilcoxon rank sum test [23] having a significance level of 0.05. Table III displays the statistical significance of the HV metric in the same way. The cases of comparing the best algorithm with itself are marked as NA (Not Applicable). The statistically equivalent performance of all algorithms is taken into account by the null hypothesis. In the event that the rank-sum p-value is less than the significance level when comparing any two algorithms, the null hypothesis is rejected. Here, 0.05 is chosen as the significance level.

Table II shows that, in 12 out of 21 cases, the suggested MaOABC algorithm performs better than the competing algorithms. Three of the twelve cases have performance differences between the suggested algorithm and the nearest competitor that are deemed negligible. For DTLZ2, MaOABC performs marginally better than GrEA when N = 6 and 8. When N=10, the same finding is made regarding MaODE and HypE for DTLZ6. When N = 10, 8, and 6, GrEA performs better than MaOABC for DTLZ1, DTLZ3, and DTLZ4. For DTLZ6 when N = 6 and 8, and for DTLZ7 when N = 6, MOEA/D is found to be better than MaOABC.Table II displays the good performance of HypE for DTLZ5.

Table III demonstrates how well MaOABC performs in achieving high HV metric values for WFG test instances. Out of 27 situations, MaOABC performs better than its competitors in 14 cases. With regard to WFG1, it can be seen that, at N=6, MOEA/D performs marginally worse than MaOABC, and at N=8, HypE performs marginally worse than MaOABC. When N = 10, the suggested algorithm MaOABC slightly outperforms the MOEA/D and HypE algorithms for WFG2. When N = 10 for WFG1 and N = 8 and 10 for WFG3, MOEA/D performs better than MaOABC. HypE also marginally outshines MaOABC for WFG2 when N = 8. For the WFG test suite, GrEA is able to achieve the second rank in obtaining near-optimal HV values, as shown in Table III.Friedman and the Iman-Davenport tests [23] are used to determine the statistical significance of the difference between the mean value of the HV and IGD metrics.

The results are shown in Tables II and III. The ranking determined by the Friedman test in relation to the HV and IGD metrics is shown in Tables IV and V, respectively. According to the null hypothesis, the ranks of all the competitor algorithms are similar because they all perform similarly. The null hypothesis is rejected because there is a significant difference in the HV and IGD metric standards attained by the contestant algorithm. This can be observed in Table IV and Table V, where it is shown that the statistical measure is greater than the critical value mentioned in brackets.

5. Conclusion

The research suggested a brand-new method for applying ABC to solve MaOO. In the paper, all the different objectives of the MaOO problem are parallelly optimized, and the solutions are subsequently combined. Next, the set of high-quality solutions formed by the union is filtered. This paper proposes a unique approach to rank the union set solutions as an alternative to the standard Pareto optimality.

In two examination suites (DTLZ and WFG), experiments show that the suggested MaOABC outperforms other algorithms in a statistically substantial way, taking into consideration the two well-known performance metrics. The conclusion is reached by applying the Friedman, Iman-Davenport, and Wilcoxon rank-sum tests.

References

[1] Monalisa Pal, Sriparna Saha, and Sanghamitra Bandyopadhyay, "Clustering Based Online Automatic Objective Reduction to Aid Many-Objective Optimization," *2016 IEEE Congress on Evolutionary Computation (CEC)*, Vancouver, BC, Canada, pp. 1131-1138, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[2] Monalisa Pal, and Sanghamitra Bandyopadhyay, "Many-Objective Feature Selection for Motor Imagery EEG Signals Using Differential Evolution and Support Vector Machine," *2016 International Conference on Microelectronics, Computing and Communications (MicroCom)*, Durgapur, India, pp. 1-6, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[3] André Sülflow, Nicole Drechsler, and Rolf Drechsler, "Robust Multi-Objective Optimization in High Dimensional Spaces," *Evolutionary Multi-Criterion Optimization: 4th International Conference*, Matsushima, Japan, pp. 715-726, 2007. [CrossRef] [Google Scholar] [Publisher Link]

[4] Peter J. Fleming, Robin C. Purshouse, and Robert J. Lygoe, "Many-Objective Optimization: An Engineering Design Perspective," *Evolutionary Multi-Criterion Optimization: Third International Conference*, Guanajuato, Mexico, pp. 14-32, 2005. [CrossRef] [Google Scholar] [Publisher Link]

[5] J. David Schaffer, *Multiple Objective Optimizations with Vector Evaluated Genetic Algorithms*, 1st ed., Proceedings of the First International Conference of Genetic Algorithms and Their Application, pp. 93-100, 1985. [Google Scholar] [Publisher Link]

[6] K. Deb et al., "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182-197, 2002. [CrossRef] [Google Scholar] [Publisher Link]

[7] Dimo Brockhoff, and Eckart Zitzler, "Are All Objectives Necessary? On Dimensionality Reduction in Evolutionary Multiobjective Optimization," *Parallel Problem Solving from Nature - PPSN IX: 9th International Conference*, Reykjavik, Iceland, pp. 533-542, 2006. [CrossRef] [Google Scholar] [Publisher Link]

[8] Yifan Li, Hai-Lin Liu, and Fangqing Gu, "An Objective Reduction Algorithm Based on Hyperplane Approximation for Many-Objective Optimization Problems," *2016 IEEE Congress on Evolutionary Computation*, Vancouver, BC, Canada, pp. 2470-2476, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[9] Yuan Yuan et al., "Objective Reduction in Many-Objective Optimization: Evolutionary Multiobjective Approaches and Comprehensive Analysis," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 2, pp. 189-210, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[10] Marco Laumanns et al., "Combining Convergence and Diversity in Evolutionary Multiobjective Optimization," *Evolutionary Computation*, vol. 10, no. 3, pp. 263-282, 2002. [CrossRef] [Google Scholar] [Publisher Link]

[11] Mario Köppen, and Kaori Yoshida, "Substitute Distance Assignments in NSGA-II for Handling Many-Objective Optimization Problems," *Evolutionary Multi-Criterion Optimization: 4th International Conference*, Matsushima, Japan, pp. 727-741, 2007, [CrossRef] [Google Scholar] [Publisher Link]

[12] Antonio López Jaimes et al., "Adaptive Objective Space Partitioning Using Conflict Information for Many Objective Optimization," *Evolutionary Multi-Criterion Optimization: 6th International Conference*, Ouro Preto, Brazil, pp. 151-165, 2011. [CrossRef] [Google Scholar] [Publisher Link]

[13] Saku Kukkonen, and Jouni Lampinen, "Ranking-Dominance and Many-Objective Optimization," *2007 IEEE Congress on Evolutionary Computation*, Singapore, pp. 3983-3990, 2007. [CrossRef] [Google Scholar] [Publisher Link]

[14] Miqing Li et al., "Enhancing Diversity for Average Ranking Method in Evolutionary Many-Objective Optimization," *Parallel Problem Solving from Nature: 11th International Conference*, Krakov, Poland, pp. 647-656, 2010. [CrossRef] [Google Scholar] [Publisher Link]

[15] L. While et al., "A Faster Algorithm for Calculating Hypervolume," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 1, pp. 29-38, 2006. [CrossRef] [Google Scholar] [Publisher Link]

[16] Johannes Bader, and Eckart Zitzler, "HypE: An Algorithm for Fast Hypervolume Based Many-Objective Optimization," *Evolutionary Computation*, vol. 19, no. 1, pp. 45-76, 2011. [CrossRef] [Google Scholar] [Publisher Link]

[17] Qingfu Zhang, and Hui Li, "MOEA/D: A Multi-Objective Evolutionary Algorithm Based on Decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712-731, 2007. [CrossRef] [Google Scholar] [Publisher Link]

[18] Shengxiang Yang et al., "A Grid-Based Evolutionary Algorithm for Many-Objective Optimization," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 5, pp. 721-736, 2013. [CrossRef] [Google Scholar] [Publisher Link]

[19] Kalyanmoy Deb et al., *Scalable Test Problems for Evolutionary Multiobjective Optimization*, Evolutionary Multiobjective Optimization, Advanced Information and Knowledge Processing, pp. 105-145, 2005. [CrossRef] [Google Scholar] [Publisher Link]

[20] S. Huband et al., "A Review of Multiobjective Test Problems and a Scalable Test Problem Toolkit," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 5, pp. 477-506, 2006. [CrossRef] [Google Scholar] [Publisher Link]

[21] Xingyi Zhang, Ye Tian, and Yaochu Jin, "A Knee Point Driven Evolutionary Algorithm for Many-Objective Optimization," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 6, pp. 761-776, 2012. [CrossRef] [Google Scholar] [Publisher Link]

[22] Miqing Li, and Jinhua Zheng, "Spread Assessment for Evolutionary Multiobjective Optimization," *Evolutionary Multi-Criterion Optimization: 5th International Conference*, Nantes, France, pp. 216-230, 2009. [CrossRef] [Google Scholar] [Publisher Link]

[23] Joaquín Derrac et al., "A Practical Tutorial on the Use of Nonparametric Statistical Tests as a Methodology for Comparing Evolutionary and Swarm Intelligence Algorithms," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 3-18, 2011, [CrossRef] [Google Scholar] [Publisher Link]

[24] Kalyanmoy Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*, Wiley, pp. 1-497, 2001. [Google Scholar] [Publisher Link]