

Migration in Multi-Population Differential Evolution for Many Objective Optimization

Pratyusha Rakshit

Basque Center for Applied Mathematics, Bilbao, Spain
Department of Electronics and Telecommunication Engineering
Jadavpur University, India
pratyushar1@gmail.com

Archana Chowdhury

Department of Computer Science Engineering
Christian College of Engineering, Bhilai, India
chowdhuryarchana@gmail.com

Amit Konar

Department of Electronics and Telecommunication Engineering
Jadavpur University, Kolkata, India
konaramit@yahoo.co.in

Atulya K. Nagar

Department of Mathematics and Computer Science, Liverpool Hope University, United Kingdom
nagara@hope.ac.uk

Abstract— The paper proposes a novel extension of many objective optimization using differential evolution (MaODE). MaODE solves a many objective optimization (MaOO) problem by parallel optimization of individual objectives. MaODE involves N populations, each created for an objective to be optimized using MaODE. The only mode of knowledge transfer among populations in MaODE is the modified version of mutation policy of DE, where every member of the population during mutation is influenced by the best members of all the populations under consideration. The present work aims at further increasing the communication between the members of the population by communicating between a superior and an inferior population, using a novel migration strategy. The proposed migration policy enables poor members of an inferior population to evolve with a superior population. Simultaneously, members from the superior population are also transferred to the inferior one to help it improving its performance. Experiments undertaken reveal that the proposed extended version of MaODE significantly outperforms its counterpart and the state-of-the-art techniques.

Keywords—differential evolution; many-objective optimization; individual parallel optimization; multiple population; migration.

I. INTRODUCTION

Many-objective optimization (MaOO) refers to optimization problems, involving four or more conflicting objectives [1]. Over the past decade, popularity of MaOO algorithms have radically increased to solve real world optimization problems, like brain-computer interfacing [2], aerospace engineering [3], extraction of bioactive compound [4], robotics [5] and many more applications. However, there exists vast literature revealing the incapability of evolutionary multi-objective optimization (EMOO) algorithms to solve MaOO problem. The non-dominance criteria used in EMOO to form Pareto fronts fail to maintain sufficient selection pressure for a MaOO problem with four or more objectives. This is due to remarkable growth in the number of Pareto optimal solutions with the increase in the number of objectives beyond three [6], [7].

Researchers are taking keen interest to overcome the impasse of EMOO to solve MaOO problems. Strategies proposed in [8], [9] aim at judicious selection of the significant objectives out of all objectives for a given MaOO

problem. Traces of reference point based MaOO algorithms are found in [10], [11] where decision maker determines the search direction in the preferred regions of the Pareto front. Among the methods, enhancing Pareto optimality criteria, to solve MaOO problems, ϵ -dominance [12], fuzzy Pareto dominance [13] and subspace dominance comparison [14] need special mentioning. The authors' previous work, referred to as *many-objective optimization using differential evolution* (MaODE) [15], utilizes the implicit parallelism of a population-based evolutionary algorithm (realized with DE) to effectively expedite the convergence of the algorithm.

MaODE solves a MaOO problem in two steps. First, individual N objectives of the given MaOO problem are optimized in parallel by N evolving populations of solutions. Second, the well-performing solutions of all N populations are combined and ranked with respect to individual objectives. The set of equally good solutions with equal *sum of rank (SoR)* measures are declared as the solution to the MaOO problem. The scope of knowledge transfer between the populations is provided by modified version of DE/current-to-best/1 mutation policy [16-19] of MaODE. Unlike the traditional DE/current-to-best/1 mutation policy, a target vector belonging to a population \mathbf{P}^k is influenced by the best members of all N populations (instead of the best candidate of \mathbf{P}^k only) to generate its donor vector.

This strategy, however, ignores the average and poor performing members of a population, which if evolved with other populations, may have excelled in their performance. The present work aims at migration of members between different populations to aid an individual to evolve with respect to an objective on which it is most likely to perform well. The migration policy first judiciously selects the communicating pair of populations from N populations. The proposed selection strategy bypasses need of any network topology. The stratagem allows the best population to communicate the worst population, the second best to the second worst, and the third best to the third worst and so on. This communication/migration policy among diverse populations enhances the exploration capability of the MaODE algorithm. The quality of a population is determined based on the median and the interquartile range (IQR) of the objective function values of its members. A population with

low median value (for minimization problem) and a high IQR is recorded as a superior population preserving diversity.

Next the identified population-pairs participate in the migration process by exchanging their selected members. An inferior population \mathbf{P}^j sends its poor-performing members to a superior population \mathbf{P}^i . This provides an opportunity to the immigrating solutions to improve their quality by evolving with \mathbf{P}^i . In return, \mathbf{P}^i sends its randomly selected members to \mathbf{P}^j . However, special care is taken to ensure that \mathbf{P}^i should not lose its well-performing members to improve the performance of the inferior populations.

The proposed MaOO algorithm, hereafter referred to as *extended* MaODE, is compared with its counterpart and three state-of-the-art techniques [20-22] to optimize the well-known DTLZ [23] and WFG [24] test suits. Experiments with different number of objectives reveal that the *extended* MaODE outperforms most of its contenders in a statistically significant manner.

The paper is divided into five sections. Sections II and III respectively overview the DE and MaODE algorithms. Section IV provides the migration policy embedded in MaODE. Experimental settings for the benchmarks and simulation strategies are explained in section V. Section VI concludes the paper.

II. DIFFERENTIAL EVOLUTION

Differential Evolution (DE) [16-19] is a population-based meta-heuristic algorithm. We now present an overview of the main steps of the DE algorithm for a minimization problem.

1. Initialization: A population $\mathbf{P}(G) = \{\vec{Z}_1(G), \vec{Z}_2(G), \dots, \vec{Z}_{NP}(G)\}$ of NP , D -dimensional real-valued target vectors is initialized randomly within the prescribed search-bound $[\vec{Z}^{\min}, \vec{Z}^{\max}]$ at generation $G=0$. The crossover rate CR is initialized in $[0, 1]$. The objective function value $f(\vec{Z}_i(0))$ is evaluated for the target vector $\vec{Z}_i(0)$ with $i = 1, 2, \dots, NP$.

2. Mutation: DE/current-to-best/1 mutation strategy is employed to generate a donor vector $\vec{Y}_i(G)$ corresponding to each $\vec{Z}_i(G)$ using (1) for $i = 1, 2, \dots, NP$.

$$\vec{Y}_i(G) = \vec{Z}_i(G) + F_1(\vec{Z}^{best}(G) - \vec{Z}_i(G)) + F_2(\vec{Z}_{r_1}(G) - \vec{Z}_{r_2}(G)) \quad (1)$$

Here $\vec{Z}^{best}(G)$ represents the best solution of the G -th generation and $\vec{Z}_{r_1}(G)$ and $\vec{Z}_{r_2}(G)$ are randomly selected population members from $\mathbf{P}(G)$ excluding $\vec{Z}_i(G)$. F_1 and F_2 are scale factors in $[0, 2]$ controlling the exploitation and the exploration capabilities respectively.

3. Crossover: The binomial crossover is concerned with selecting components from $\vec{Z}_i(G)$ and $\vec{Y}_i(G)$ to generate a trial vector $\vec{W}_i(G)$. The selection of components from $\vec{Z}_i(G)$ and $\vec{Y}_i(G)$ is controlled by the crossover ratio CR as follows.

$$w_{i,j}(G) = \begin{cases} y_{i,j}(G), & \text{if } \text{rand} \leq CR \text{ or } j = j_{rand} \\ z_{i,j}(G), & \text{otherwise} \end{cases} \quad \text{for } j = 1, 2, \dots, D \quad (2)$$

with rand as a uniformly distributed random number in $(0, 1)$. Here j_{rand} represents a randomly chosen index from $\{1, 2, \dots, D\}$. The objective function value $f(\vec{W}_i(G))$ is evaluated for $\vec{W}_i(G)$. This is repeated for $i = 1, 2, \dots, NP$.

4. Selection: A greedy selection process promotes the fitter candidate among $\vec{Z}_i(G)$ and $\vec{W}_i(G)$ to the next generation as described by (3) for $i = 1, 2, \dots, NP$.

$$\vec{Z}_i(G+1) = \begin{cases} \vec{W}_i(G), & \text{if } f(\vec{W}_i(G)) \leq f(\vec{Z}_i(G)) \\ \vec{Z}_i(G), & \text{otherwise} \end{cases} \quad (3)$$

5. Convergence: After each evolution cycle, steps 2 to 4 are reiterated until terminating criterion is satisfied.

III. OVERVIEW OF DIFFERENTIAL EVOLUTION INDUCED MANY-OBJECTIVE OPTIMIZATION (MaODE)

MaODE, proposed in [15], comprises two primary steps. First, each of the N objectives of the MaOO problem is handled by an individual DE. All N DEs are run in parallel to simultaneously optimize all N objectives. The second step deals with judiciously identify the common pool of solutions, which maximally satisfy all N objectives.

Parallel optimization of N objectives: The crux of the first step is the premise that a single solution of a MaOO problem may not optimize all $N > 4$ objectives adequately. Hence it is desirable to optimize all N objectives in parallel and individually with an aim to obtain the pool of optimal solutions corresponding to each objective. This optimization policy envisages capturing the characteristics of all $N > 4$ objectives effectively. The individual DE algorithm handling optimization of the specific k -th objective is referred to as k -MDE (*modified* DE) for $k = 1, 2, \dots, N$. The steps of k -MDE are similar to the traditional DE, except for two modifications, as illustrated next.

Let \vec{Z}^{k-best} with the minimum objective function value $f_k(\vec{Z}^{k-best})$ be the fittest candidate evolved by k -MDE after attaining convergence for $k \in \{1, 2, \dots, N\}$. In the present context of MaOO, \vec{Z}^{k-best} may not be the optimal solution for the l -th objective, where $l \in \{1, 2, \dots, N\}$ but $l \neq k$. Contrarily, a solution of k -MDE, which is slightly poor to \vec{Z}^{k-best} (with fitness say $(1+\alpha) \times f_k(\vec{Z}^{k-best})$ for a small positive constant α), may emerge as a potential candidate for the remaining one or more objectives. In other words, only keeping the best solutions \vec{Z}^{k-best} of all N k -MDEs for $k = 1, 2, \dots, N$ may lead to dismissal of promising regions in the search space jointly satisfying all objectives. Two approaches are proposed to overcome this impasse.

First, after attaining convergence, we save a group of best-fit solutions Ψ_k of k -MDE, instead of \vec{Z}^{k-best} only for $k \in \{1, 2, \dots, N\}$. The range of fitness of the solutions to enter Ψ_k is set as $[f_k(\vec{Z}^{k-best}), (1+\alpha) \times f_k(\vec{Z}^{k-best})]$. It allows ingress of the top $100 \times (1-\alpha)\%$ best-fit solutions with respect to the k -th

objective only for $k \in \{1, 2, \dots, N\}$. Evidently, a setting of $\alpha = 0$ allows only \bar{Z}^{k-best} in Ψ_k . More solutions are allowed to occupy Ψ_k with an increase in α . Mathematically,

$$\Psi_k = \left\{ \bar{Z} \mid f_k(\bar{Z}) \in \text{range}_k \text{ and } \bar{Z} \in \bigcup_{G=1}^{G_{\max}} \mathbf{P}^k(G) \right\}. \quad (4.a)$$

$$\text{with } \text{range}_k = \left[f_k(\bar{Z}^{k-best}), (1+\alpha) \times f_k(\bar{Z}^{k-best}) \right] \quad (4.b)$$

with $\mathbf{P}^k(G)$ denoting the population of k -MDE at the G -th generation.

Second, to enjoy the composite benefits of the promising regions explored by all N optimization problems, the DE/current-to-best/1 mutation policy is amended as follows.

$$\bar{Y}_i^k(G) = \bar{Z}_i^k(G) + \sum_{l=1}^N F_l^1 (\bar{Z}^{l-best}(G) - \bar{Z}_i^k(G)) + F_2 (\bar{Z}_{r_1}^k(G) - \bar{Z}_{r_2}^k(G)) \quad (5)$$

Here the second term allows the i -th individual $\bar{Z}_i^k(G)$ of $\mathbf{P}^k(G)$ to gather knowledge from the best candidates $\bar{Z}^{l-best}(G)$ (for $l = 1, 2, \dots, N$) of all N optimization problems at generation G . $\bar{Z}_{r_1}^k(G)$ and $\bar{Z}_{r_2}^k(G)$ are randomly selected members of $\mathbf{P}^k(G)$ excluding $\bar{Z}_i^k(G)$.

We employ N such k -MDEs in parallel, each dealing with optimization of the k -th objective only for $k = \{1, 2, \dots, N\}$.

After obtaining the pool of best fit solutions Ψ_k (corresponding to the k -th objective) for $k = 1, 2, \dots, N$, we take their union given by

$$\Omega = \Psi_1 \cup \Psi_2 \cup \dots \cup \Psi_N. \quad (6)$$

Evidently, each members of Ω satisfy at least one objective among N objectives to an extent of $100 \times (1-\alpha)\%$.

Parallel optimization of N objectives: The second step of MaODE is concerned with identifying the pool of equally good solutions from Ω that maximally optimize all N objectives. This is accomplished in two phases. In the first phase, the rank of each member of Ω is evaluated with respect to individual N objectives. In the later phase, the optimal set of solutions (of the MaOO problem) with equal *sum of ranks* (*SoRs*) is identified.

For the first phase, each of the N objective function values of all the solutions of Ω is evaluated. In other words, all the remaining $N-1$ objective functions $f_l(\cdot)$ are evaluated for the members of Ψ_k (and Ω as well) with already evaluated k -th objective $f_k(\cdot)$ for $k, l = 1, 2, \dots, N$ but $k \neq l$. Let f_k^{best} be the best fitness value of the k -th objective after evaluating $f_k(\cdot)$ for all members of Ω for $k = 1, 2, \dots, N$. A solution $\bar{Z} \in \Omega$ is assigned with a rank 1 with respect to the k -th objective if its fitness $f_k(\bar{Z}) \in [f_k^{best}, (1+\alpha) \times f_k^{best}]$ for $k \in \{1, 2, \dots, N\}$. This signifies that the members acquiring rank 1 with respect to the specific k -th objective optimizes $f_k(\cdot)$ by $100 \times (1-\alpha)\%$ or more. If $f_k(\bar{Z}) \in [(1+\alpha) \times f_k^{best}, (1+2\alpha) \times f_k^{best}]$, \bar{Z} is assigned with a rank 2 with respect to the k -th objective $f_k(\cdot)$. Similarly, the set of solutions of rank 3 with respect to $f_k(\cdot)$ is identified from Ω

with fitness in $[(1+2\alpha) \times f_k^{best}, (1+3\alpha) \times f_k^{best}]$. Symbolically, a solution $\bar{Z} \in \Omega$ is assigned with a rank $r_k(\bar{Z}) = R$ with respect to $f_k(\cdot)$ if

$$f_k(\bar{Z}) \in \left[(1+(R-1)\alpha) f_k^{best}, (1+R\alpha) f_k^{best} \right]. \quad (7)$$

This rank evaluation is continued for $k = 1, 2, \dots, N$ for all members of Ω . Thus, the *SoR* of $\bar{Z} \in \Omega$ is obtained by

$$\text{SoR}(\bar{Z}) = \sum_{k=1}^N r_k(\bar{Z}). \quad (8)$$

It is evident from the above description that the solutions achieving rank 1 for all N objectives have $\text{SoR} = N \times 1 = N$. The solutions which obtain a rank 2 for a specific objective, but rank 1 for all the remaining $N-1$ objectives, are assigned with $\text{SoR} = (N-1) \times 1 + 2 = N+1$ and so on. Smaller the *SoR* better is the quality of a solution.

After evaluating *SoR* for all members of Ω , the second phase commences with an aim to identify a subset $\Omega_1 \subset \Omega$, consisting of the members of Ω with $\text{SoR} = N$. In other words, Ω_1 includes the candidates of Ω which acquire rank 1 with respect to all N objectives. If no such Ω_1 can be identified, the strategy searches for another subset $\Omega_2 \subset \Omega$ including the members with $\text{SoR} = N+1$. This process is continued until a non-empty subset Ω_l is identified from Ω with its constituent members of $\text{SoR} = N+l-1$. The Ω_l is declared as the optimal set of solutions Ω^{opt} to the given MaOO problem.

IV. PROPOSED EXTENSION OF MaODE

The aim of the MaODE is to allow a solution to evolve with respect to a specific objective only, which it may proficiently optimize. However, MaODE suffers from a constraint of inadequate communication between multiple DE populations, each concerned with optimization of specific objective. The only mode of communication employed in MaODE is to utilize the amended version of DE/current-to-best/1 mutation policy, given in (5). In the modified mutation strategy, a target vector $\bar{Z}_i^k \in \mathbf{P}^k$ is attracted to the best solution $\bar{Z}^{l-best} \in \mathbf{P}^l$ of each population (for $l = 1, 2, \dots, N$) found so far. This in turn allows \bar{Z}_i^k to communicate only with the best candidates \bar{Z}^{l-best} evolved by N k -MDE algorithms for $k, l = 1, 2, \dots, N$.

However, it is intuitive that apart from the best solution, the remaining population members evolved by a k -MDE for $k \in \{1, 2, \dots, N\}$ may appear to be significant candidates for the remaining objectives too. Moreover, only $100 \times (1-\alpha)\%$ best-fit solutions of Ψ_k for $k = 1, 2, \dots, N$ participate in forming Ω , from where ultimately the optimal set of solutions of MaOO problem is obtained. The strategy thus ignores the average and poor performing solutions of k -MDE (for $k \in \{1, 2, \dots, N\}$) which if evolved, by one of the remaining $N-1$ DEs, may emerge as potential candidates for the given MaOO problem. This insists for a suitable method of migration of population members across different fitness landscapes to transfer domain knowledge effectively. This in turn helps in exploring more potential regions in the parameter space with respect to all N objectives.

The proposed migration strategy is discussed next.

1. First, the members of each population \mathbf{P}^k are arranged in ascending order of their objective function values $f_k(\cdot)$ (for minimization problem) for $k = 1, 2, \dots, N$.
2. Each population \mathbf{P}^k is then represented by the median f_k^{med} and the interquartile range (IQR) ϑ_k of the sorted objective function values of its constituent members for $k = 1, 2, \dots, N$.
3. The N populations are first arranged in the ascending order of their median fitness values f_k^{med} and the rank thus obtained by \mathbf{P}^k in the sorted list is denoted by $rank_k^1$ for $k = 1, 2, \dots, N$. Evidently, the fittest population acquires rank 1.
4. The same set of N populations are again sorted in the descending order of their respective IQR of fitness measures ϑ_k and the rank acquired by \mathbf{P}^k in this sorted list is represented by $rank_k^2$ for $k = 1, 2, \dots, N$. The population occupying the first position preserves the maximum diversity (required to ensure the exploration capability).
5. The composite rank of \mathbf{P}^k is thus obtained by

$$rank_k = rank_k^1 + rank_k^2 \quad (9)$$

for $k = 1, 2, \dots, N$.

6. The N populations are now sorted in the ascending order of their respective composite ranks $rank_k$ for $k = 1, 2, \dots, N$. It is noteworthy that the population occupying the first position now can be treated as the best-performing population with quality fitness measure and population diversity.
7. This step is concerned with identifying the emigration-immigration population-pair based on the composite ranks of N populations. The premise is based on the principle of *fair division* [25], where the migration takes place between the superior and the inferior populations (instead of random pair or based on any specific network topology). The quality of the populations is determined based on their respective composite ranks. The fair division method employs an organized migration policy between the first $N/2$ populations and the remaining $N/2$ populations (which are already sorted in ascending order of their composite ranks $rank$). Here, the migration occurs between the best population and the worst population, the second best and the second worst, and the third best and the third worst and so on. The cooperation between two vastly different populations, targets to improve the population diversity and the possibility of exploration as well.
8. Once the pairs of communicating populations are determined, the next task is to identify the migrating population members. Let \mathbf{P}^i - \mathbf{P}^j be a population-pair participating in the migration process with \mathbf{P}^i (and \mathbf{P}^j) as the superior (and the inferior) population for $i, j = 1, 2, \dots, N$ but $i \neq j$. The migration is undertaken in two phases.
 - (a) First, the proposed approach gives an opportunity to the poor members of an inferior population \mathbf{P}^j to improve their quality by migrating to a better population \mathbf{P}^i . The members of \mathbf{P}^j are already sorted in ascending order of their objective function values (as in step-1). Let n be the number of candidates of \mathbf{P}^j that possess objective function

values greater than the upper quartile of the sorted fitness measures (for minimization problem). Evidently, these are the poor-performing members of \mathbf{P}^j . They are given an opportunity to improve their performance by evolving with better population. So this set of n poor-performing solutions from an inferior population \mathbf{P}^j is selected for migration to a better population \mathbf{P}^i .

- (b) However, to keep the size of each population fixed at NP , the strategy also migrates n members of \mathbf{P}^i to \mathbf{P}^j . The n solutions are chosen randomly from the members of the sorted population \mathbf{P}^i (as done in step-1), which possess objective function values above the lower quartile. This selection policy ensures that the population \mathbf{P}^i should not lose its quality members with objective function values less than the lower quartile (for minimization problem). It is based on the supposition that it is better to preserve a quality solution in a population rather than eliminating it from the given population to enhance the performance of other populations.

The migration strategy outlined above is undertaken after every G_M generations of k -MDE for $k = 1, 2, \dots, N$. The amended versions of k -MDE and MaODE to solve MaOO problem are referred to as *extended k-MDE* and *extended MaODE* henceforth. The pseudo-codes are given below.

Procedure extended k -MDE

Input: Population size NP , Crossover rate CR , Optimizing objective function $f_k(\cdot)$, Migrating generation gap G_M .

Output: Set of solutions ψ_k with objective measures in $[f_k^{best}, (1+\alpha)f_k^{best}]$.

Begin

I. Initialization:

- (i) Set the generation number $G \leftarrow 0$ and randomly initialize D -dimensional NP target vectors $\bar{Z}_i^k(G)$ of population $\mathbf{P}^k(G)$ for $i = 1, 2, \dots, NP$.
- (ii) Evaluate the objective function value $f_k(\bar{Z}_i^k(G))$ for $i = 1, 2, \dots, NP$.
- (iii) Set $\bar{Z}^{k-best}(G) \leftarrow \arg \left(\min_{i=1}^{NP} (f_k(\bar{Z}_i^k(G))) \right)$.
- (iv) Set $range_k = \left[f_k(\bar{Z}^{k-best}(G)), (1+\alpha) \times f_k(\bar{Z}^{k-best}(G)) \right]$ and

$$S(G) = \left\{ \bar{Z} \mid f_k(\bar{Z}) \in range_k \text{ and } \bar{Z} \in \mathbf{P}^k(G) \right\}.$$

II. While stopping criterion is not reached, do begin

- (a) **Migration:** If $\text{mod}(G, G_M) = 0$ then do
Call **migration**($\mathbf{P}^1(G), \mathbf{P}^2(G), \dots, \mathbf{P}^N(G)$) to get updated population $\mathbf{P}^k(G)$.

End If.

- (b) **Mutation:** Generate a donor vector $\bar{Y}_i^k(G)$ corresponding to the candidate solution $\bar{Z}_i^k(G)$ using (5) for $i = 1, 2, \dots, NP$.
- (c) **Crossover:** Generate a trial vector $\bar{W}_i^k(G)$ corresponding to the candidate solution $\bar{Z}_i^k(G)$ and the donor vector $\bar{Y}_i^k(G)$ using (2) $i = 1, 2, \dots, NP$.

- (d) Evaluate the objective function value $f_k(\vec{W}_i^k(G))$ for $i = 1, 2, \dots, NP$.
- (e) Set $\vec{Z}_i^k(G+1) \leftarrow \vec{W}_i^k(G)$ if $f_k(\vec{W}_i^k(G)) \leq f_k(\vec{Z}_i^k(G))$ for $i = 1, 2, \dots, NP$.
- (f) Set $\vec{Z}^{k-best}(G+1) \leftarrow \vec{W}_i^k(G)$ if $f_k(\vec{W}_i^k(G)) \leq f_k(\vec{Z}^{k-best}(G))$.
- (g) Set $range_k = \left[f_k(\vec{Z}^{k-best}(G+1)), (1+\alpha) \times f_k(\vec{Z}^{k-best}(G+1)) \right]$ and $S(G+1) \leftarrow \left\{ \vec{Z} \mid f_k(\vec{Z}) \in range_k \text{ and } \vec{Z} \in \mathbf{P}^k(G+1) \cup S(G) \right\}$.
- (h) Increase the generation value $G \leftarrow G+1$.
- End While.**
- III. Set $\psi_k \leftarrow S(G)$.
- IV. Return ψ_k .
- End.**

Procedure migration

Input: N populations $\mathbf{P}^1(G), \mathbf{P}^2(G), \dots, \mathbf{P}^N(G)$.

Output: Updated populations after migration.

Begin

1. Sort members of \mathbf{P}^k in ascending order of $f_k(\cdot)$ for $k = 1, 2, \dots, N$.
2. Sort \mathbf{P}^k s in ascending order of its median objective function value f_k^{med} and identify its rank $rank_k^1$ for $k = 1, 2, \dots, N$.
3. Sort \mathbf{P}^k s in descending order of its IQR g_k of objective function values and identify its rank $rank_k^2$ for $k = 1, 2, \dots, N$.
4. Determine the composite rank $rank_k$ using (9) for $k = 1, 2, \dots, N$.
5. Sort \mathbf{P}^k s in ascending order of $rank_k$ for $k = 1, 2, \dots, N$ and let the sorted populations be $\mathbf{P}^{1-sort}(G), \mathbf{P}^{2-sort}(G), \dots, \mathbf{P}^{N-sort}(G)$.
6. **For** $i = 1:N/2$
 - (a) Set $\mathbf{P}^i \leftarrow \mathbf{P}^{i-sort}$ and $\mathbf{P}^j \leftarrow \mathbf{P}^{(N-i+1)-sort}$.
 - (b) Select the set of members ξ^j of \mathbf{P}^j with objective function values greater than the upper quartile of \mathbf{P}^i .
 - (c) Set $n \leftarrow |\xi^j|$.
 - (d) Randomly select n members ξ^i from \mathbf{P}^i with objective function values greater than the lower quartile of \mathbf{P}^j .
 - (e) Exchange members of ξ^i and ξ^j .

End For.

7. Return the updated populations to the corresponding optimization problem.

End.

Procedure extended_MaODE

Input: N objectives of MaOO problem.

Output: Approximate global optima of the MaOO problem.

Begin

1. Run N , **extended_k-MDE**($NP, CR, f_k(\cdot)$) in parallel to obtain ψ_k for $k = 1, 2, \dots, N$.
2. Set $\Omega \leftarrow \psi_1 \cup \psi_2 \cup \dots \cup \psi_N$.
3. Evaluate $f_k(\cdot)$ for $k = 1, 2, \dots, N$ for all members of Ω .
4. Evaluate $r_k(\vec{Z})$ for $k = 1, 2, \dots, N$ using (7) and thus $SoR(\vec{Z})$ using (8) for all $\vec{Z} \in \Omega$.
5. Set $l \leftarrow 1$.
6. Identify the solutions of Ω , with $SoR=N$ and include them in Ω_l .
7. **While** Ω_l is empty
 - Set $l \leftarrow l+1$.
 - Identify the solutions of Ω , with $SoR=N+l-1$ and include them in Ω_l .
8. Set $\Omega^{app} \leftarrow$ the first non-empty subset Ω_l of Ω with equally good trial solutions with minimum $SoRs$.

9. Return Ω^{app} .

End.

V. EXPERIMENTAL RESULTS

A. Comparative Framework and Parameter Setting

The proposed algorithm *extended* MaODE is compared with its counterpart MaODE [15] and three state-of-the-art MaOO techniques including *hypervolume estimation algorithm for multi objective optimization* (HypE) [20], *multi objective evolutionary algorithm based on decomposition* (MOEA/D) [21] and *grid based evolutionary algorithm* (GrEA) [22].

Parameter settings of MaODE and extended MaODE: Like MaODE, the performance of the *extended* MaODE also depends on prudent selection of α , CR , F_1 and F_2 . A setting of $\alpha = 0$ allows only the best solution with respect to each of the N objectives to enter Ω . It leads to loss of diversity. Contrarily, a large value of α may allow remarkable increase in the number of poor solutions in Ω degrading the performance of the algorithm. Thus, an exhaustive experiment is undertaken by varying α from 1 to 0 (with a subtractive step of 0.005). A favorable balance between the quality performance and the diversity is obtained for $\alpha = 0.05$. It allows 95% best-fit solutions of each of the N objectives to enter Ω . The crossover ratio CR and scale factors F_1 and F_2 for the proposed algorithm and its counterpart are varied within the range of (0, 1) to check the appropriate value to be set so as to get quality performance (with respect to performance metrics discussed next). Since the performance does not change significantly for $CR \geq 0.9$, $F_1 \geq 0.5$ and $F_2 \geq 0.5$, we set $CR = 0.9$, $F_1 = 0.5$ and $F_2 = 0.5$.

Reproduction operator parameters: For GrEA, MOEA/D and HyPE, the control parameters for simulated binary crossover (SBX) and polynomial mutation are set as 20 each. The crossover probability (for SBX policy) and the mutation probability are respectively set as 1 and $1/D$ for D dimensional search space.

Grid divisions: The grid divisions of GrEA for different test functions are set following the guidelines in [22].

Neighborhood size: The neighborhood size for MOEA/D is set as 20.

Sampling size: The hypervolume in HypE is approximated by the Monte Carlo method. The corresponding sampling size is set to 10,000 as given in [20].

Population size: The population size of GrEA and HyPE are set as the population size of MOEA/D depending on the number of reference points and the number of objectives. For $N = 6, 8$ and 10 , the population size T is respectively set as 132, 156 and 275 [21]. For MaODE and its extended version, the population size NP for each objective is set as T/N .

B. Benchmark Functions and Terminating Condition

Two benchmark suites, namely DLTZ [23] and WFG [24], are used to analyze the performance of the contender algorithms. The DTLZ and WFG suits respectively comprise of nine and seven benchmark functions. For each benchmark function, the number of objectives N takes values from $\{6, 8, 10\}$.

The number of variables D for DTLZ test suite is set to $N+k-1$. The parameter k is set to 5 and 20 for DTLZ1 and DTLZ7 respectively. The remaining five benchmark problems of the DTLZ test suit consider $k = 10$. For the WFG test suite, D is set to $K + L$. The distance parameter L is set as 10 for all the WFG benchmark instances irrespective of the number of objectives N . However, the position parameter K takes values of 10, 7, and 9 for $N = 6, 8,$ and 10 respectively [15].

The terminating condition for the contender algorithms is specified by the maximum number of generations. For DTLZ1 and WFG2, the maximum number of generations is fixed at 700, while for DTLZ3 and WFG1, it is set as 1000. For the remaining ones, it is limited to 250.

C. Performance Metrics

The comparative analysis of the performance of the contender algorithms is undertaken with respect to two performance metrics, including the *hypervolume* (HV) [26] and the *inverted generational distance* (IGD) [27]. Let ξ^* be the set of points distributed uniformly along the optimal Pareto front and ξ^{app} be the approximate Pareto front obtained by a MaOO algorithm, both in the objective space. It is to be noted that in case of extended MaODE and its counterpart, ξ^{app} denotes the objective function values of the members of Ω^{app} .

(a) Hypervolume: The total size of the objective space, dominated by the members of ξ^{app} represents its HV . A reference point representing the *worst possible point* or the *anti-optimal* point [28] is used to evaluate HV . After normalizing objective function values of the solutions in ξ^{app} in $(0, 1)$, the reference point is selected as $(1, 1, 1, \dots, 1)$. To employ Monte-Carlo approach [29] to evaluate HV , a set of 10^6 sampling points are considered in the hyper-rectangle between the origin and the reference point in the objective space. The fraction of the sample points, dominated by the members of ξ^{app} , inside the hyper-rectangle represents the hypervolume $HV(\xi^{app})$. An attainment function is defined for all the sampled points u in the hyper-rectangle and hypervolume is obtained by averaging the attainment function values of the sample points inside the hyper-rectangle. The attainment function is given by

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

A higher value of HV measure is an indicator of better performance of the given MaOO algorithm.

(b) Inverted Generational Distance: IGD [27] considers the minimum Euclidean distance $d(u, \xi^{app})$, between the points in $u \in \xi^*$ and points in ξ^{app} . A set of 500 uniformly sampled points is considered to represent ξ^* . The performance of a MaOO algorithm is better, if the points in ξ^{app} are close to the points in ξ^* . Thus a low value of IGD is an indicator of good performance.

D. Performance Analysis

The proposed algorithm is compared with the counterpart algorithm MaODE and other three contenders HypE, MOEA/D and GrEA with respect to the median and interquartile range (IQR) of the performance metrics (HV and IGD) over 50 independent runs. Wilcoxon rank sum test [30] is used to analyze the statistical significance of 50 sample

values of the performance metrics. However, the Wilcoxon test is undertaken between the best algorithm (providing near optimal value of the performance metric) and each of the remaining algorithms with a significance level of 0.05. The null hypothesis of the statistical test assumes all algorithms being compared are statistically equivalent. The null hypothesis is rejected if the p -value of the Wilcoxon rank sum test is found to be less than 0.05. Tables I and II respectively report the median and the IQR (within parenthesis) of HV and IGD metric values obtained by the contender algorithms over 50 independent runs respectively. The p -values obtained by the Wilcoxon test is given under brackets. The p -values are marked as NA i.e. *not applicable* when the best algorithm is compared with itself. The best results are reported in bold in Tables I and II.

The results reported in Table-I reveal that extended MaODE algorithm supersedes MaODE in 24 cases out of 27 benchmark instances with respect to HV metric. Among these 24 instances, four cases are found where the performances of extended MaODE and its counterpart are not statistically significant. These four cases include WFG2 with $N = 8$, WFG3 with $N = 6$ and 8 and WFG4 with $N = 10$. Out of the remaining 3 cases, it can be observed that for WFG8 and $N = 8$, the performance of MaODE and the extended MaODE is similar, however, the extended MaODE achieves smaller IQR. For the rest 2 cases of WFG1 with $N = 6$ and WFG5 with $N = 8$, MaODE insignificantly outperforms the extended MaODE with respective p -values of 0.05674 and 0.05518. It substantiates the merit of the proposed migration policy to enhance the performance of MaODE. MaODE achieves the second rank among the contenders. GrEA supersedes the extended MaODE in cases of WFG4 with $N = 6$, WFG6 with $N = 6$ and WFG 8 with $N = 10$.

Similar trend of performance of the competitor algorithms is observed from Table-II with respect to IGD metric. The extended MaODE achieves the best rank for 18 cases out of 21 instances of DTLZ test suite. Here also MaODE emerges as the second best algorithm by superseding HypE, MOEA/D and GrEA) in 13 cases.

TABLE-I-A: PERFORMANCE EVALUATION WITH RESPECT TO HV FOR WFG1 AND WFG2

Func.	N	HypE	MOEA/D	GrEA	MaODE	Extended MaODE
WFG1	6	9.218e-01 (9.004e-03) [1.345e-02]	9.677e-01 (2.602e-02) [5.827e-02]	9.304e-01 (3.901e-02) [4.587e-02]	9.858e-01 (2.454e-02) [NA]	9.784e-01 (2.987e-02) [5.674e-02]
	8	9.161e-01 (4.489e-02) [5.433e-02]	9.045e-01 (4.749e-02) [2.437e-02]	9.040e-01 (5.703e-02) [1.881e-02]	9.173e-01 (4.084e-02) [2.395e-02]	9.792e-01 (3.734e-02) [NA]
	10	9.108e-01 (1.148e-01) [2.733e-02]	9.488e-01 (1.568e-02) [2.386e-02]	8.493e-01 (3.669e-02) [3.222e-02]	8.444e-01 (5.768e-02) [3.416e-02]	9.749e-01 (2.693e-02) [NA]
WFG2	6	6.676e-01 (9.680e-02) [3.238e-02]	8.495e-01 (8.527e-02) [3.395e-02]	9.348e-01 (7.229e-02) [4.725e-02]	9.689e-01 (2.554e-02) [3.781e-02]	9.982e-01 (2.678e-02) [NA]
	8	9.920e-01 (2.610e-03) [2.297e-02]	9.710e-01 (2.178e-02) [1.044e-02]	9.414e-01 (9.894e-02) [1.181e-02]	9.841e-01 (3.091e-03) [5.546e-02]	9.981e-01 (2.916e-03) [NA]
	10	9.878e-01 (2.983e-03) [5.309e-02]	9.734e-01 (3.971e-02) [5.036e-02]	9.572e-01 (1.780e-02) [2.293e-02]	9.949e-01 (1.965e-03) [3.472e-02]	9.991e-01 (1.984e-03) [NA]

TABLE-I-B: PERFORMANCE EVALUATION WITH RESPECT TO *HV* FOR WFG3 TO WFG9

Func.	N	HypE	MOEA/D	GrEA	MaODE	Extended MaODE
WFG3	6	5.420e-01 (7.047e-03) [2.080e-02]	4.827e-01 (6.446e-03) [3.310e-02]	5.881e-01 (8.292e-03) [NA]	5.707e-01 (7.371e-03) [5.851e-02]	6.256e-01 (7.983e-03) [NA]
	8	5.933e-01 (1.739e-02) [6.067e-02]	5.938e-01 (5.275e-03) [NA]	5.730e-01 (1.316e-02) [4.164e-02]	5.901e-01 (2.399e-02) [5.209e-02]	6.429e-01 (2.689e-02) [NA]
	10	5.538e-01 (4.830e-03) [2.703e-02]	6.018e-01 (2.150e-03) [NA]	4.966e-01 (1.616e-02) [4.349e-02]	5.708e-01 (3.535e-03) [4.911e-02]	6.361e-01 (2.946e-03) [NA]
WFG4	6	7.495e-01 (2.298e-02) [1.590e-02]	7.516e-01 (1.347e-02) [3.227e-02]	8.026e-01 (1.071e-02) [NA]	7.871e-01 (1.639e-02) [4.699e-02]	7.973e-01 (1.587e-02) [4.485e-02]
	8	7.296e-01 (3.030e-02) [2.723e-02]	6.535e-01 (1.976e-02) [3.196e-02]	7.879e-01 (8.003e-03) [3.236e-02]	8.401e-01 (6.694e-03) [3.648e-02]	8.853e-01 (6.937e-03) [NA]
	10	5.988e-01 (5.458e-02) [2.719e-02]	4.787e-01 (1.629e-02) [3.605e-02]	8.114e-01 (8.871e-03) [4.169e-02]	7.992e-01 (1.117e-02) [5.168e-02]	8.417e-01 (2.439e-02) [NA]
WFG5	6	6.498e-01 (1.668e-02) [5.290e-03]	5.315e-01 (1.187e-02) [5.485e-03]	7.035e-01 (5.934e-03) [2.583e-02]	7.016e-01 (7.392e-03) [1.093e-02]	7.375e-01 (6.162e-03) [NA]
	8	5.657e-01 (1.317e-02) [2.022e-02]	4.125e-01 (1.185e-02) [2.241e-02]	6.572e-01 (5.914e-03) [3.817e-02]	7.137e-01 (6.247e-03) [NA]	7.014e-01 (5.671e-03) [5.518e-02]
	10	3.760e-01 (1.535e-02) [3.859e-02]	3.333e-01 (8.460e-03) [4.664e-02]	5.843e-01 (5.784e-03) [4.863e-02]	6.103e-01 (4.789e-03) [3.762e-02]	6.683e-01 (5.267e-03) [NA]
WFG6	6	4.585e-01 (7.447e-03) [6.944e-03]	6.557e-01 (3.900e-02) [9.601e-03]	6.945e-01 (2.380e-02) [NA]	5.906e-01 (2.068e-02) [4.681e-02]	6.527e-01 (2.613e-02) [5.576e-02]
	8	2.620e-01 (1.833e-02) [2.627e-02]	4.920e-01 (2.282e-02) [2.651e-02]	6.157e-01 (1.590e-02) [4.305e-02]	6.938e-01 (1.423e-02) [2.568e-02]	7.615e-01 (1.892e-02) [NA]
	10	3.969e-01 (1.068e-02) [2.600e-02]	4.593e-01 (3.319e-02) [3.357e-02]	7.704e-01 (1.930e-02) [5.706e-02]	7.861e-01 (1.028e-02) [4.782e-02]	8.376e-01 (2.998e-02) [NA]
WFG7	6	3.656e-01 (1.657e-02) [1.310e-02]	5.924e-01 (3.111e-02) [1.738e-02]	6.521e-01 (2.662e-02) [2.930e-02]	6.766e-01 (7.862e-03) [1.2391e-02]	7.268e-01 (6.625e-03) [NA]
	8	6.242e-01 (4.410e-02) [1.213e-02]	5.496e-01 (3.474e-02) [2.212e-02]	7.367e-01 (8.555e-03) [3.774e-02]	7.806e-01 (8.214e-03) [3.152e-02]	8.317e-01 (7.142e-03) [NA]
	10	6.300e-01 (3.549e-02) [1.973e-02]	3.069e-01 (8.094e-03) [3.439e-02]	8.625e-01 (7.448e-03) [3.681e-02]	8.747e-01 (6.463e-03) [2.968e-02]	9.037e-01 (6.962e-03) [NA]
WFG8	6	3.776e-01 (1.519e-02) [3.417e-02]	2.065e-01 (1.978e-02) [2.211e-02]	4.481e-01 (3.052e-02) [3.056e-02]	4.346e-01 (2.565e-02) [3.520e-02]	4.947e-01 (2.831e-02) [NA]
	8	2.746e-01 (1.708e-02) [1.351e-02]	3.129e-01 (2.249e-02) [1.654e-02]	4.322e-01 (1.650e-02) [2.121e-02]	6.056e-01 (1.962e-02) [NA]	6.056e-01 (1.571e-02) [NA]
	10	3.001e-01 (2.038e-02) [2.149e-02]	5.299e-01 (2.620e-02) [4.108e-02]	7.937e-01 (3.604e-02) [NA]	5.722e-01 (1.512e-02) [4.431e-02]	6.923e-01 (1.613e-02) [4.632e-02]
WFG9	6	3.436e-01 (3.511e-02) [3.775e-02]	3.935e-01 (4.076e-02) [3.845e-02]	7.097e-01 (4.352e-02) [4.173e-02]	6.968e-01 (4.450e-02) [4.042e-02]	7.863e-01 (4.653e-02) [NA]
	8	4.126e-01 (6.416e-02) [1.887e-02]	4.880e-01 (2.691e-02) [3.952e-02]	5.650e-01 (1.504e-02) [4.746e-02]	6.269e-01 (1.1027e-02) [3.472e-02]	6.915e-01 (1.792e-02) [NA]
	10	4.348e-01 (3.707e-02) [2.193e-02]	5.563e-01 (3.626e-02) [3.356e-02]	7.640e-01 (6.116e-03) [NA]	6.856e-01 (5.906e-03) [4.167e-02]	7.561e-01 (6.124e-03) [4.261e-02]

TABLE-II: PERFORMANCE EVALUATION WITH RESPECT TO *IGD* FOR DTLZ TEST SUITE

Func.	N	HypE	MOEA/D	GrEA	MaODE	Extended MaODE
DTLZ 1	6	2.292e-01 (4.357e-02) [1.173e-02]	1.085e-01 (5.856e-02) [2.876e-02]	1.373e-01 (6.745e-03) [1.765e-02]	9.918e-02 (3.776e-03) [2.876e-02]	9.131e-02 (3.366e-03) [NA]
	8	3.148e-01 (2.534e-02) [2.151e-03]	1.866e-01 (1.377e-02) [8.450e-03]	2.511e-01 (3.665e-02) [4.106e-02]	1.349e-01 (1.035e-02) [3.524e-02]	1.142e-01 (1.956e-02) [NA]
	10	2.378e-01 (3.667e-02) [3.238e-02]	2.218e-01 (3.333e-02) [3.245e-02]	1.395e-01 (1.786e-02) [4.5856e-02]	1.422e-01 (2.887e-02) [5.688e-02]	1.226e-01 (2.385e-02) [NA]
DTLZ 2	6	4.697e-01 (5.360e-02) [1.481e-02]	4.066e-01 (4.701e-02) [2.735e-02]	2.574e-01 (1.894e-03) [2.735e-02]	2.571e-01 (1.662e-03) [4.621e-02]	2.012e-01 (1.724e-03) [NA]
	8	7.017e-01 (4.494e-02) [1.842e-02]	6.644e-01 (5.700e-02) [3.128e-02]	3.669e-01 (3.383e-03) [5.033e-02]	3.535e-01 (3.551e-03) [4.672e-02]	3.157e-01 (3.813e-03) [NA]
	10	7.329e-01 (4.447e-02) [3.901e-02]	7.497e-01 (6.853e-02) [3.878e-02]	3.950e-01 (2.600e-02) [4.646e-02]	1.421e-01 (2.224e-02) [4.783e-02]	1.161e-01 (2.548e-02) [NA]
DTLZ 3	6	4.487e-01 (7.654e-02) [2.079e-02]	4.937e-01 (1.910e-01) [2.233e-02]	2.167e-01 (6.232e-02) [3.471e-02]	2.167e-01 (7.235e-02) [3.582e-02]	2.592e-01 (7.625e-02) [NA]
	8	6.344e-01 (1.227e-01) [2.553e-02]	7.754e-01 (1.287e-01) [3.974e-02]	3.778e-01 (1.982e-01) [4.627e-02]	4.244e-01 (4.167e-02) [4.088e-02]	3.673e-01 (4.725e-02) [NA]
	10	7.369e-01 (5.950e-02) [2.664e-02]	6.874e-01 (2.717e-01) [3.221e-02]	4.959e-01 (1.517e-01) [4.057e-02]	3.971e-01 (7.456e-02) [3.705e-02]	3.387e-01 (7.582e-02) [NA]
DTLZ 4	6	6.383e-01 (1.209e-01) [2.750e-02]	5.268e-01 (1.679e-02) [4.379e-02]	2.658e-01 (2.210e-03) [3.278e-02]	2.678e-01 (1.442e-03) [5.695e-02]	2.385e-01 (1.521e-03) [NA]
	8	7.033e-01 (7.503e-02) [3.112e-02]	7.596e-01 (6.516e-02) [1.506e-02]	3.783e-01 (2.881e-03) [4.935e-02]	3.521e-01 (2.780e-03) [3.682e-02]	3.146e-01 (2.537e-03) [NA]
	10	8.256e-01 (1.769e-02) [1.152e-02]	7.810e-01 (2.023e-02) [2.354e-02]	4.040e-01 (9.373e-03) [4.221e-02]	3.572e-01 (1.327e-02) [4.527e-02]	3.216e-01 (1.638e-02) [NA]
DTLZ 5	6	1.530e-02 (9.717e-03) [NA]	2.412e-02 (1.539e-02) [3.138e-02]	8.920e-02 (1.586e-02) [1.129e-02]	7.449e-02 (1.458e-02) [4.178e-02]	5.599e-02 (1.746e-02) [3.836e-02]
	8	5.020e-02 (1.022e-02) [2.151e-02]	8.441e-02 (1.922e-02) [1.555e-02]	1.703e-01 (4.758e-02) [4.616e-02]	3.213e-02 (3.367e-02) [3.572e-02]	3.042e-02 (3.749e-02) [NA]
	10	2.934e-02 (2.917e-03) [2.418e-02]	7.163e-02 (4.168e-02) [4.898e-02]	4.155e-01 (7.877e-02) [2.194e-02]	2.969e-01 (5.794e-02) [6.024e-02]	2.614e-01 (5.582e-02) [NA]
DTLZ 6	6	3.557e-01 (5.688e-02) [1.311e-02]	1.587e-01 (3.154e-01) [NA]	2.520e-01 (6.178e-02) [2.043e-02]	2.055e-01 (1.544e-01) [2.970e-02]	1.652e-01 (1.739e-01) [2.361e-02]
	8	4.522e-01 (3.403e-01) [3.014e-02]	1.658e-01 (2.918e-02) [2.583e-02]	5.590e-01 (1.735e-01) [1.108e-02]	2.227e-01 (4.116e-02) [3.556e-02]	1.521e-01 (4.469e-02) [NA]
	10	2.320e-01 (4.720e-02) [5.593e-02]	4.624e-01 (5.534e-02) [2.539e-02]	4.116e-01 (2.717e-01) [2.120e-02]	2.288e-01 (4.076e-02) [2.683e-02]	2.173e-01 (4.628e-02) [NA]
DTLZ 7	6	1.555e+00 (1.276e-01) [4.276e-03]	4.024e-01 (1.652e-02) [NA]	5.602e-01 (2.501e-02) [4.005e-02]	5.772e-01 (4.624e-03) [1.312e-02]	5.143e-01 (4.634e-03) [1.548e-02]
	8	2.831e+00 (2.222e-01) [2.892e-02]	1.342e+00 (2.423e-01) [3.651e-02]	1.001e+00 (2.334e-02) [4.644e-02]	9.327e-01 (1.963e-02) [3.547e-02]	9.051e-01 (1.468e-02) [NA]
	10	2.960e+00 (1.684e-01) [2.294e-02]	1.543e+00 (4.554e-01) [2.734e-02]	1.321e+00 (9.439e-02) [4.815e-02]	1.216e+00 (8.852e-02) [3.891e-02]	1.172e+00 (8.529e-02) [NA]

VI. CONCLUSION

The paper proposed a novel extension of the authors' previous work on MaODE. The primary objective of MaODE is to optimize individual N objectives (of a MaOO problem) in parallel so as to preserve the significance of each objective and helping the solutions to evolve in a specific fitness landscape where they perform well. The present work enhances the performance of MaODE by devising a novel migration policy to exchange members of two populations, each concerned with optimization of a specific objective function. The migration strategy allows a poor population to receive members from a well-performing population. It in turn helps to improve the performance of the poor population. Simultaneously, the inferior members of the poor population are transferred to the well-performing population. This gives an opportunity to these inferior members to evolve with a well-performing population for possible enhancement of their quality. Experimental results reveal a statistically significant quality performance of the proposed MaOO algorithm over its contenders.

REFERENCES

- [1] H. Ishibuchi, N. Tsukamoto, and Y. Nojima, "Evolutionary many-objective optimization: A short review," in *IEEE Congress on Evolutionary Computation*, 2008, pp. 2419-2426.
- [2] M. Pal and S. Bandyopadhyay, "Many-objective feature selection for motor imagery EEG signals using differential evolution and support vector machine," in *IEEE International Conference on Microelectronics, Computing and Communication, MicroCom*, 2016, pp. 1-6.
- [3] P. J. Fleming, R. C. Purshouse, and R. J. Lygoe, "Many-objective optimization: an engineering design perspective," in *International Conference, EMO*, 2005, pp. 14-32.
- [4] K. D. Shashi, V. K. Katiyar, "Multi objective extraction optimization of bioactive compounds from gardenia using real coded genetic algorithm," in *World Congress of Biomechanics*, 2010, pp. 1463-1466.
- [5] Y. Fang, Q. Liu, M. Li, Y. Laili, and D. T. Pham, "Evolutionary many-objective optimization for mixed-model disassembly line balancing with multi-robotic workstations," *European Journal of Operational Research*, vol. 276, no. 1, 2019, pp. 160-174.
- [6] B. Cao, J. Zhao, Z. Lv, X. Liu, S. Yang, X. Kang, and K. Kang, "Distributed parallel particle swarm optimization for multi-objective and many-objective large-scale optimization," *IEEE Access*, vol. 5, 2017, pp. 8214-8221.
- [7] W. Hu, G. G. Yen, and G. Luo, "Many-objective particle swarm optimization using two-stage strategy and parallel cell coordinate system," *IEEE transactions on cybernetics*, vol. 47, no. 6, 2016, pp. 1446-1459.
- [8] D. Brockhoff and E. Zitzler, "Are all objectives necessary? On dimensionality reduction in evolutionary multiobjective optimization," in *Parallel Problem Solving from Nature*, 2006, pp. 533-542.
- [9] H. K. Singh, A. Isaacs, and T. Ray, "A Pareto corner search evolutionary algorithm and dimensionality reduction in many-objective optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 4, 2011, pp. 539-556.
- [10] K. Deb and J. Sundar, "Preference point based multi-objective optimization using evolutionary algorithms," in *Genetic and Evolutionary Computation Conference*, 2006, pp. 635-642.
- [11] K. Deb, and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point based non-dominated sorting approach, Part I: solving problems with box constraints," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, 2014, pp. 577-601.
- [12] M. Laumanns, L. Thiele, K. Deb, and E. Zitzler, "Combining convergence and diversity in evolutionary multiobjective optimization," *Evolutionary Computation*, vol. 10, no. 3, 2002, pp. 263-282.
- [13] M. K"oppen and K. Yoshida, "Substitute distance assignments in NSGA-II for handling many-objective optimization problems," in *Evolutionary Multi-Criterion Optimization*, 2007, pp. 727-741.
- [14] A. L. Jaimes, C. A. Coello Coello, H. Aguirre, and K. Tanaka, "Adaptive objective space partitioning using conflict information for many objective optimization," in *Evolutionary Multi-Criterion Optimization*, 2011, pp. 151-165.
- [15] P. Rakshit, A. Chowdhury, A. Konar, and A. K. Nagar, "Differential evolution induced many objective optimization," in *IEEE Congress on Evolutionary Computation*, 2017, pp. 2240-2247.
- [16] R. Storn, and K. Price, "Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, 1997, pp. 341-359.
- [17] S. Das, and P. N. Suganthan, "Differential evolution: a survey of the state-of-the-art," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, 2011, pp. 4-31.
- [18] P. Rakshit, A. Konar, P. Bhowmik, I. Goswami, S. Das, L. C. Jain, and A. K. Nagar, "Realization of an adaptive memetic algorithm using differential evolution and Q-learning: a case study in multirobot path planning," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 43, no. 4, 2013, pp. 814-831.
- [19] P. Rakshit, A. Konar, S. Das, L. C. Jain, and A. K. Nagar, "Uncertainty management in differential evolution induced multiobjective optimization in presence of measurement noise," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 44, no. 7, 2014, pp. 922-937.
- [20] J. Bader and E. Zitzler, "HypE: an algorithm for fast hypervolume based many-objective optimization," *Evolutionary Computation*, vol. 19, no. 1, 2011, pp. 45-76.
- [21] Q. Zhang and H. Li, "MOEA/D: a multi-objective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, 2007, pp. 712-731.
- [22] S. Yang, M. Li, X. Liu, and J. Zheng, "A grid-based evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 5, 2013, pp. 721-736.
- [23] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable test problems for evolutionary multiobjective optimization," in *Evolutionary Multiobjective Optimization: Theoretical Advances and Applications*, A. Abraham, L. Jain, and R. Goldberg, Eds., Springer, 2005, pp. 105-145.
- [24] S. Huband, P. Hingston, L. Barone, and L. While, "A review of multiobjective test problems and a scalable test problem toolkit," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 5, 2006, pp. 477-506.
- [25] H. Moulin, *Fair division and collective welfare*, MIT press, 2004.
- [26] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*, New York: Wiley, 2001.
- [27] M. Li and J. Zheng, "Spread assessment for evolutionary multiobjective optimization," in *5th International Conference on Evolutionary Multi-Criterion Optimization*, 2009, pp. 216-230.
- [28] L. While, P. Hingston, L. Barone, and S. Huband, "A Faster Algorithm for Calculating Hypervolume," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 1, 2006, pp. 29-38.
- [29] X. Zhang, Y. Tian, and Y. Jin, "A knee point driven evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 6, 2012, pp. 761-776.
- [30] J. Derrac, S. García, D. Molina, and F. Herrera, "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, 2011, pp. 3-18.