

An enhanced bacterial colony optimization with dynamic multi-leader co-evolution for multiobjective optimization problems

Hong Wang¹ | Yixin Wang¹ | Menglong Liu² | Tianwei Zhou¹ | Ben Niu¹

¹College of Management, Shenzhen University, Shenzhen, China

²School of Mechanical Engineering and Automation, Harbin Institute of Technology (Shenzhen), Shenzhen, China

Correspondence

Ben Niu, College of Management, Shenzhen University, Shenzhen, China.

Email: drniuben@gmail.com

Funding information

National Natural Science Foundation of China, Grant/Award Numbers: 71971143, 71901152; Social Science Youth Foundation of Ministry of Education of China, Grant/Award Number: 21YJC630181; Guangdong Basic and Applied Basic Research Foundation, Grant/Award Numbers: 2023A1515011032, 2023A1515010919; Guangdong Innovation Team Project of Intelligent Management and Cross Innovation, Grant/Award Number: 2021WCXTD002; Shenzhen Stable Support Grant, Grant/Award Number: 20220810100952001; Special Projects in Key Fields of Universities in Guangdong Province, Grant/Award Number: 2022ZDZX2054; Guangdong Provincial Philosophy and Social Sciences Planning Project, Grant/Award Number: GD22XGL22

Abstract

The information transfer mechanism within the population is an essential factor for population-based multiobjective optimization algorithms. An efficient leader selection strategy can effectively help the population to approach the true Pareto front. However, traditional population-based multiobjective optimization algorithms are restricted to a single global leader and cannot transfer information efficiently. To overcome those limitations, in this paper, a multiobjective bacterial colony optimization with dynamic multi-leader co-evolution (MBCO/DML) is proposed, and a novel information transfer mechanism is developed within the group for adaptive evolution. Specifically, to enhance convergence and diversity, a multi-leaders learning mechanism is designed based on a dynamically evolving elite archive via direction-based hierarchical clustering. Finally, adaptive bacterial elimination is proposed to enable bacteria to escape from the local Pareto front according to convergence status. The results of numerical experiments show the superiority of the proposed algorithm in comparison with related population-based multiobjective optimization algorithms on 24 frequently used benchmarks. This paper demonstrates the effectiveness of our dynamic leader selection in information transfer for improving both convergence and diversity to solve multiobjective optimization problems, which plays a significant role in information transfer of population evolution. Furthermore, we confirm the validity of the co-evolution framework to the bacterial-based optimization algorithm, greatly enhancing the searching capability for bacterial colony.

KEY WORDS

bacterial colony optimization, dynamic multi-leader learning, elite co-evolution, evolutionary direction, hierarchical clustering, population-based multiobjective optimization

1 | INTRODUCTION

Many applications in real life often contain multiple objectives to be optimized simultaneously such as in the fields of intelligent manufacture systems, environment energy systems, financial and management science, and so forth, which are termed multiobjective optimization problems (MOPs) (Liu et al., 2022; Ma et al., 2023; Morteza et al., 2023; Nunes et al., 2023; Yazdani et al., 2023). For example, the portfolio optimization problem is to minimize the risk and maximize the return (Morteza et al., 2023). In the routing planning problem, along with the goal of shortest route distance, it is also vital to consider cost, safety, and risk (Nunes et al., 2023). Frequently, multiple objectives in MOPs are contradictory to each other. Thus, MOPs are usually solved by finding a set of trade-off solutions known as the Pareto optimal set rather than a unique global optimal solution.

Compared to the single objective optimization problem, MOPs are more challenging in terms of computational complexity, convergence difficulties, the balance between the convergence and diversity of Pareto solutions, and so forth. Since population-based metaheuristic can produce a set of Pareto optimal solutions in a single run, they are more suitable for solving the complex MOPs. Those population-based multiobjective optimization algorithms, for example, multiobjective particle swarm optimization (MOPSO) (Aboud et al., 2022; Liu et al., 2019), multiobjective ant colony optimization (MOACO) (Li et al., 2022; Zhang et al., 2021), non-dominated sorting genetic algorithm (NSGAII) (Deb, Agrawal, et al., 2002), and multiobjective differential evolution (Liao et al., 2023), have excellent global search capabilities, primarily in identifying key knowledge in generations, designing efficient learning methods, and incorporating some heuristics to explore potential subspaces and ultimately achieving global optimization. More specifically, the population-based algorithms operate on a population of solutions, called individuals. The information transfer mechanism among the individuals is an essential factor for population-based multiobjective optimization algorithms to achieve better solutions iteratively, for example, the interaction between individual and the global optimal (Lin et al., 2018; Pereira & Gomes, 2023), the topology exchanges among the whole population (Niu et al., 2018; Niu, Liu, & Tan, 2019).

Though population-based multiobjective algorithms have excellent global search capabilities that naturally have the advantage of tackling MOPs, there are still several challenges during implementations. On the one hand, it is known that an efficient population-based multiobjective algorithm needs to design rational mechanisms to transfer the historic knowledge from the individuals for the movement of the remaining population. Generally, it is required to select a global best for updating and reproducing the next circulation in a population-based algorithm (He et al., 2022; Tan et al., 2017; Zhang et al., 2019). Prior knowledge acts as an intermediary in the group and plays a role in directing the group towards the potential space. Even so, it is not that easy to design a rational and efficient information transfer mechanism within the group (Wu et al., 2019; Zhang et al., 2019). The main challenge is to specify if it is the personal best or the global best for the whole population since it is difficult to select an absolute optimum for the MOPs with conflicting goals. Applying the idea of multi-population for multiobjective is a potential solution for this issue (Li et al., 2022; Yang et al., 2022). The multi-population mechanism not only reduces computation complexity by assigning each subpopulation with a relatively easier subtask, but it is also capable of improving overall robustness of the entire population to avoid trapping in local Pareto optimal (Antonio & Coello, 2018). On the other hand, the balance between the convergence and diversity in the population is a tough challenge throughout the entire optimization process (Gu et al., 2023; Yang et al., 2023). Fast convergence with the cost of rapid loss of diversity may result in the evolution trapping in the local optimum. Thus, selecting an appropriate evolutionary direction, that is, selecting appropriate leaders for each individual in population, as well as the strategy for controlling the balance between convergence and diversity are the key aspects for applying population-based optimization methods to the MOPs. Informed by these two motivations, we endeavour to select the most suitable multiple leaders, that is, the convergence leader and the diversity leader, in dynamic subpopulations in order to achieve a balance between convergence and diversity in MOPs.

Bacterial-based optimization is a type of novel population-based algorithm, which simulates the foraging behaviours and life-cycle of bacterial colony (Niu & Wang, 2012; Passino, 2012). First, as one of the oldest species, the survival of bacteria demonstrates a great capacity for optimization. The simple and accessible bionic mechanisms make it easy to grasp the optimization mechanism. Second, different from the existing population-based optimization algorithms, bacterial-based optimization itself has excellent adaptive operators, that is, the elimination process can help the whole population jump out of the difficult space and avoid trapping in stagnation long time. Lastly, due to its explicit global search and local search mechanisms, we can easily extend this optimization structure to match different engineering problems. Benefit to the excellent global search capability and simple bionic structure, bacterial-based optimization algorithms have been widely studied and successfully applied in many multiobjective optimization engineering fields (Guo, Tang, Niu, & Lee, 2021). As a population-based algorithm, the main effort of the multi-objective bacterial-based optimization algorithms (MOBAs) is to find out the most advanced prior information within the group to guide the updating of the entire population. To the best of our knowledge, the searching structure improvements (Niu et al., 2018; Niu, Liu, & Tan, 2019), learning strategies (Niu, Yi, et al., 2019; Tan et al., 2017), and searching efficiency improvements (Guo, Tang, & Niu, 2021; Niu et al., 2020) are most widely investigated to improve the efficiency of the MOBAs. Though existing versions can enhance the performance of MOBAs in tackling the MOPs to a certain extent, inefficient transfer mechanisms of prior information, low robustness of leader selection mechanism, inappropriate use of external archival technology, and imbalance between convergence and diversity are prevalent in extant MOBAs.

Thus, in this study, we propose a novel multiobjective bacterial-based optimization algorithm, named multiobjective bacterial colony optimization with dynamic multi-leader co-evolution algorithm (abbreviated as MBCO/DML), to solve the above-mentioned problems. First, to avoid the low robustness caused by conventional single global best guidance, an efficient information transfer mechanism is proposed, that is, selecting appropriate diversity and convergence leaders in dynamic subpopulations. Our strategy dynamically divides the entire population into subpopulations according to evolutionary status during generations. Each subpopulation can maintain and retain the prior information suitable for them instead of sharing global best. Meanwhile, the balance of convergence and diversity can be achieved by learning from convergence and diversity leaders. Moreover, to accelerate the optimization process and avoid trapping into local Pareto optimal, the elite archive is also capable in evolving independently. The global information is stored in an elite archive and shared with each subpopulation, which ensures communication among subpopulations. The main contributions of this study are as follows.

- We propose a novel multiobjective bacterial colony optimization framework that a co-evolution framework is constructed and an adaptive evolutionary path is designed, which provides a new information transfer mechanism for population-based multiobjective optimization algorithms.
- The dynamic multi-leader learning mechanism driven by a direction-based hierarchical clustering algorithm is proposed to guide the whole bacterial colony. Each dynamic subpopulation can independently choose the best evolutionary direction that suits them and ultimately obtain better convergence and diversity performance.
- An elite archive evolution strategy is effectively integrated into the evolution process. We confirm the validity of the co-evolution framework to the bacterial-based optimization algorithm, not only greatly enhancing the searching capability for bacterial colony, but also can speed up the optimization process.

The reminder of this paper is organized as follows. Section 2 briefly describes the background of MOPs, and the state-of-art multiobjective bacterial-based optimization algorithms. Next, the MBCO/DML is introduced in Section 3. In Sections 4 and 5, the comparison experiments on benchmarks are conducted, followed by an analysis of results obtained. Finally, the conclusions and future work are shown in Section 6.

2 | RELATED WORK

The relevant theories will be carefully examined and presented in this section. To begin with, the theory of MOPs and Pareto optimal will be briefly introduced. Next, since the proposed MBCO/DML is based on bacterial intelligent behaviours, some bacterial based optimization works will also be reviewed.

2.1 | MOPs and Pareto optimal

Mathematically, a MOP can be defined as follow.

$$\begin{aligned} & \text{Minimize } F(x) = (f_1(x), f_2(x), f_3(x), \dots, f_m(x)) \\ & \text{Subject to } g_i(x) \leq 0, i = 1, 2, 3, \dots, q, \end{aligned} \quad (1)$$

where, x is a n -dimensional decision variable. The objective function $F(x)$ includes m objectives constrained by q constraint conditions $g(x)$. In general, it is difficult to obtain the optimal solution for each objective simultaneously. A solution may be optimal in one objective, but may not be superior in other objectives, which determines that the MOP is pursuing compromise solutions instead of just a certain optimal solution.

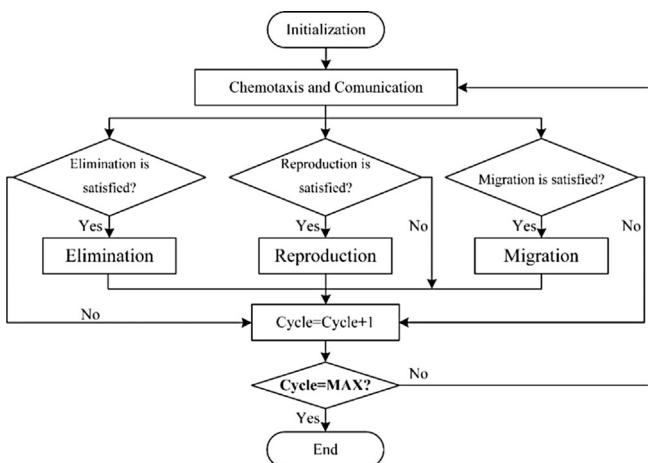
It is supposed that the optimization problem is a minimization problem. An objective value vector $F(u) = (f_1(u), f_2(u), \dots, f_m(u))$ is said to dominate to another $F(v) = (f_1(v), f_2(v), \dots, f_m(v))$ if and only if $f_i(u) \leq f_i(v), \forall i \in \{1, 2, \dots, m\}$ and $f_j(u) < f_j(v), \exists j \in \{1, 2, \dots, m\}$, denoted as $F(u) < F(v)$. When an objective value vector is not dominated by any others, its decision variable is called as a non-dominance solution, also known as Pareto optimality. For example, if no feasible solution x' can be found to satisfy $F(x) < F(x')$, the feasible solution x is called as a Pareto optimality.

2.2 | Bacterial colony optimization

Bacterial Colony Optimization (BCO) is proposed by Niu and Wang (2012) to alleviate the computational complexity as well as low convergence speed in the earliest bacterial-based optimization algorithms (e.g., Bacterial Foraging Optimization (Passino, 2012), Bacteria Chemotaxis (Muller et al., 2002)) by replacing the original triple nested loop structure to parallel structure.

As is shown in Figure 1, all parts of the bacterial behaviours in BCO are designed into a parallel structure and distributed throughout the optimization process, which simplifies the process of colony optimization. The other behaviours, such as elimination, reproduction and migration, can only be implemented if the given conditions are met. BCO with its unique and adaptive biological behaviours ensure that the colony can search for excellent solutions from a global perspective. This is especially suitable for complex optimization problems such as high-dimensional, multi-modal, non-linear and multi-constrained characteristics.

Except for the improvements in life-cycle structure, in BCO, the convergence strategy is designed based on adopting topology communication structures, like dynamic neighbour-oriented learning and group-oriented learning. Two strategies are also designed in the chemotaxis process to assume the role of optimization consisting of swimming and tumbling. To be specific, the current position is decided by the previous position and the advanced group experience is generated by the global best and individual history experience provided by the personal best. The swimming process can be described as follows.

**FIGURE 1** The framework of BCO.

Swimming:

$$p_i = p_i + r_i(gbest - p_i) + (1 - r_i)(pbest_i - p_i), \quad (2)$$

where, p_i is the position of the i th iteration; the $gbest$ and $pbest_i$ represent the global best within the population and the personal history best of the i th bacterium, respectively; the r_i is the randomly generated learning rate for the i th bacterium.

Compared to swimming, the tumbling process includes an additional randomness operation to avoid trapping into local optimum and premature. Tumbling process can be described as follows.

Tumbling:

$$p_i = p_i + r_i(gbest - p_i) + (1 - r_i)(pbest_i - p_i) + C_i \frac{\Delta_i}{\sqrt{\Delta_i^T \times \Delta_i}}, \quad (3)$$

where, C_i represents the chemotaxis step size of i th bacterium; Δ_i is the direction angle of the i th individual generated in $[-1, 1]$ randomly.

2.3 | Multiobjective bacterial-based optimization algorithms

Bacterial-based optimization algorithms have been extensively studied and extended to the multiobjective optimization field (Kaur & Kadam, 2018; Srichandan et al., 2018). Niu et al. (2013) first proposed a multiobjective bacterial foraging optimization (MBFO). They first validated that there is a great potential of applying BFO in MOPs. In MBFO, the Pareto dominance relationship and the external archive technique were first employed in conventional BFO structures, whose simulation results show that MBFO can find a much better spread of Pareto solutions and converge faster than other classical algorithms. To speed up the convergence rate, a comprehensive learning strategy is embedded in MBFO-cl proposed by Tan et al. (2017) to enable the communication exchanges between the bacteria and external archive. In Yi et al. (2016) proposed an enhanced MOBFO, in which parallel cell entropy was introduced to evaluate the evolutionary status of the Pareto solutions and an adaptive foraging strategy was applied to balance the convergence and diversity in the optimization procedure. Niu, Liu, and Tan (2019) proposed a multi-swarm cooperative multiobjective bacterial foraging algorithm (MCMBFO), which aims to accelerate the bacterial colony convergence rate by cooperation. Even the simulation results from MOBFO and MCMBFO illustrated a good performance in terms of converging to the true Pareto front, the nested loop structure was still applied in these two methods, which undoubtedly slowed down the convergence speed.

To alleviate the high-computing complexity of the standard BFO algorithm, Niu et al. (2020) redesigned the nested loop structure to a single life cycle and incorporated the ring topology search to propose a multiobjective bacterial colony optimization (abbreviated as MORBCO). MORBCO involves global chemotaxis operation, elite reproduction strategy and personal best archive with neighbourhood communication mechanism, which enhanced the local search capability of MORBCO. In Niu, Yi, et al. (2019), similar learning paradigms, including elite learning, star topology learning, ring topology learning and Von Neumann topology learning, are compared with each other and exploited in multiobjective feature selection tasks, in which experimental results demonstrated that enhanced learning strategies can improve the convergence speed and avoid

premature to some extent. For both MCMBFO (Niu, Liu, & Tan, 2019) and MORBCO (Niu et al., 2020), topological communication is employed as the main information transfer path can improve search efficiency to some extent but they neglect to identify what important information should be restored and transferred within the population for tackling MOPs. In Dhillon et al. (2016) and Wang and Cai (2018), PSO-like single global best learning paradigm was also combined with BFO to solve MOPs and applied to the load frequency control problem and crashworthiness optimization of vehicle body respectively. Guo, Tang, and Niu (2021) proposed an evolutionary state-based novel multiobjective periodic bacterial foraging optimization algorithm, named ES-NMPBFO. In ES-NMPBFO, all bacteria can adjust the direction of learning from a global leader and use a periodic learning structure that can save most of the computation resources. Although ES-NMPBFO identifies the evolutionary status to adjust the global leader can enhance the adaptive of the population, the single global best information transfer mechanism still performs not well in the balance of convergence and diversity and is easily trapped into the local optimal. From the above-mentioned literature, we believe that the leader selection strategy and key information identification and transfer in the population are decisive factors affecting the optimization efficiency.

The above-mentioned MOBAs confirm the potential of the bacterial-based optimization algorithm for multiobjective optimization. However, there are some drawbacks associated with existing MOBAs. They select their learning object randomly and exhibit blind reliance on a single global leader with low robustness. As a class of swarm intelligence, little attention has been paid to identifying key prior information in the population and designing a reasonable and effective information transfer mechanism within the group for MOBAs tackling MOPs.

Thus, in this study, a multiobjective bacterial colony optimization with dynamic multileader co-evolution algorithm (MBCO/DML) is presented, which further improves the capability of convergence and diversity, and overcomes the imbalance between them with the help of effective information transfer mechanism, robust leader selection strategy and adaptive co-evolution technique.

3 | THE PROPOSED METHODOLOGY

In this section, an improved multiobjective bacterial colony optimization, abbreviated as MBCO/DML, is proposed to enhance the tradeoff between diversity and convergence. The overall framework of the proposed algorithm is shown in Figure 2.

Specifically, clustering-based dynamic multi-leaders selection mechanism is designed to select the convergence leaders and diversity leaders which can guide the conductive searching of the optimum. An elite archive is designed to store the nondominated solutions which will be used to

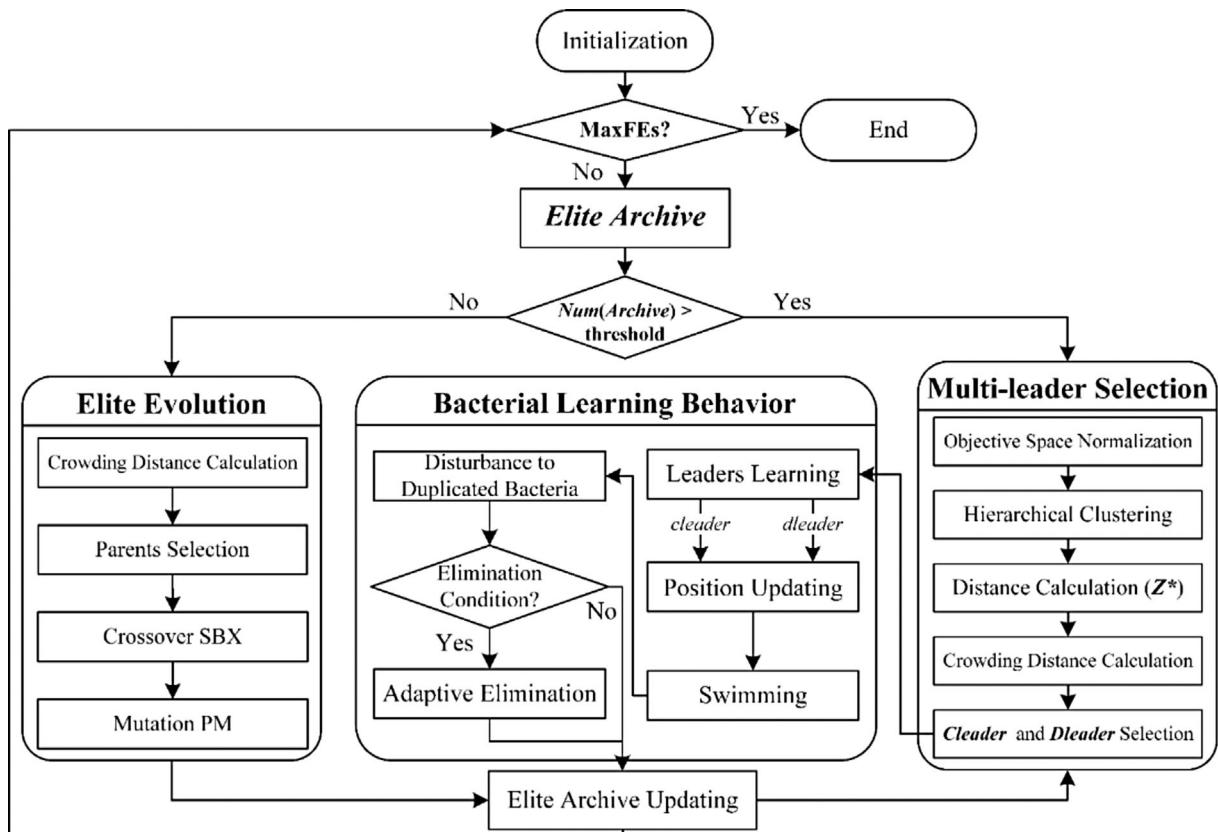


FIGURE 2 The overall framework of the proposed MBCO/DML.

form the final Pareto set. Different from other archives with a storage function only, the proposed MBCO/DML extracts all nondominated solutions from the current population to form an archive that will evolve independently using parents-selection, simulated binary crossover (SBX) and polynomial mutation. Additionally, an adaptive elimination paradigm embedded in the bacterial learning behaviour is developed to improve optimization efficiency and avoid trapping in local optimum. The key components of the proposed MBCO/DML are illustrated in detail in the following sections.

3.1 | Clustering-based dynamic multi-leader learning strategy

For most multiobjective bacterial-based optimization algorithms, the entire population is guided by a global optimum. However, if all individuals are guided in their evolution by only a global optimum, as shown in Figure 3a, we can find that the whole population can easily trap in local optimum and perform badly on diversity. The main task of MOP is to obtain solutions as many and as varied as possible. Thus, relying solely on a single global optimum to guide the movement of the entire population is inefficient and inconsistent with the reality of multiobjective optimization.

In the proposed MBCO/DML, the diversity and convergence of the population are fully considered in leadership selection. As shown in Figure 3b, the dynamic multi-leader selection mechanism is implemented by clustering the whole population into subpopulations, and the diversity leader and convergence leader are produced in each subpopulation to guide the conductive searching of remaining individuals. During the evolutionary process, each subpopulation is not stationary. The subpopulations can find their proper leader and evolution direction in a relatively small searching space by separating the population into a certain number of clusters using a hierarchical clustering method. Specifically, the similarity between pairs of bacteria in the population will be calculated according to the direction difference. Two most similar bacteria will be combined, and that combination will be conducted repeatedly until a cluster tree is produced. The bacterial colony is therefore separated into several clusters, that is, subpopulations. For each bacterial cluster, a convergence leader *cleader* and a diversity leader *dleader* are generated according to convergence and diversity measurements. The *cleader* represents the bacterium with the best convergence evaluation in each bacterial cluster, while the *dleader* indicates the one with the best diversity metric. The prior knowledge of the convergence and diversity are therefore exploited to promote the overall performance of subgroups in handling the MOPs.

For a MOP, the solution scale of each objective dimension is not at the same level. This imbalance brings some inconvenience to the calculation process and bias to the optimization objective. Therefore, the normalization operation is required to be executed at the beginning of each iteration, which is usually a compression according to the current obtained optimal and worst solutions. Moreover, the ideal point is not usually easy to obtain during the optimization process but it is an important indicator to define evolutionary status that can be employed to guide population evolution. The superior point $z^* = (z_1^*, z_2^*, \dots, z_m^*)$ of a MOP is defined by the minimum value of each objective dimension, where $z_j^* = \min_{p \in P} sf_j(p)$, $j = 1, 2, \dots, m$, p is the individual of the whole population P . Correspondingly, the inferior point $z^- = (z_1^-, z_2^-, \dots, z_m^-)$ can be calculated

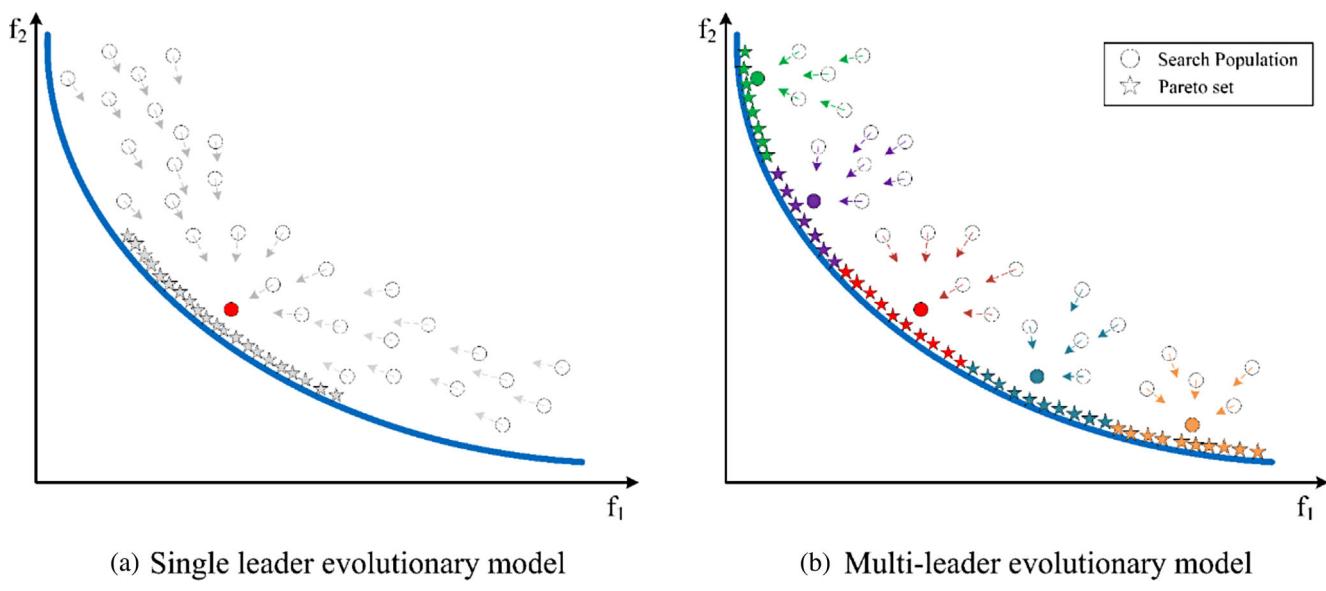


FIGURE 3 Two types of evolution models for Pareto solutions. In (a) subfigure, the red dot represents the global optimum and the grey pentagrams indicate the set of possible Pareto solutions after learning evolution. In (b) subfigure, there are several leaders for different subpopulations to guide evolution, which reproduce more diverse Pareto solutions.

by $z_j^* = \max_{p \in S} f_j(p)$, $j = 1, 2, \dots, m$. Thus, the objective space can be normalized by Equation (4) according to the superior point and the inferior point. After normalization, the superior point is further employed for leader selection and elimination.

$$f'_j(p) = \frac{f_j(p) - z_j^*}{z_j^* - z_j^*} \in [0, 1]. \quad (4)$$

To acquire bacterial colonies with similar evolutionary status, unlike the previous metric employing distance, the evolutionary direction is used to quantify similarity between bacteria and assign high similarity colonies to the same group. As is shown in Figure 4, if the position of bacterium a is kept constant and bacterium b is moved away from the origin of the coordinate axis in the original direction, then the cosine distance remains the same at this time because the movement direction and angle remain the same. In contrast, when the positions of bacteria a and c are changed, the Euclidean distance between two bacteria are keep changing. Therefore, the Euclidean distance measures the absolute distance between two points directly related to the coordinates of the location of each point, while the cosine distance measures the angle of the space vector, which is reflected in the direction difference rather than the location difference.

The pseudocode of the dynamic multi-leader selection mechanism is presented in Algorithm 1, which consists of two steps: (1) normalization and clustering and (2) leader selection.

1. Normalization and clustering

The input of dynamic multi-leader selection is Population P , Elite archive EA , predefined clustered number $cluNum$. It is noted that the elite archive is extracted from the whole population by dominant relationships. After dynamic multi-leader selection, we can get the convergence leader $cleader_c$ and diversity leader $dleader_c$ in each bacterial cluster, which represent the most suitable learning object for individuals in terms of diversity and convergence. In lines 1–2, P and EA are combined to get the union population U and obtain the best value of each objective z^* and the worst one z^- . After union, all individuals in U are normalized to $[0, 1]$ by Equation (4) to get a new normalized population $norP$ (line 3). Then, cosine similarity is applied to measure the directional similarity of every two bacteria in $norP$, getting the similarity measure $cosSim(a, b)$ using Equation (5) that represents the cosine value of two bacteria (lines 4–8).

$$cosSim(a, b) = \frac{\sum_{j=1}^m a_j b_j}{\sqrt{\sum_{j=1}^m a_j^2} \sqrt{\sum_{j=1}^m b_j^2}}. \quad (5)$$

It is noted that the more similar the directions of the two vectors are, the angle of two vectors is, and the cosine value is closer to zero. Based on the hierarchical clustering method, as seen in Figure 5, the entire population is clustered into several subpopulation according to their directions. No matter whether it is searching population or elite archive, they are assigned with cluster labels showing that they have the same evolutionary direction (line 9).

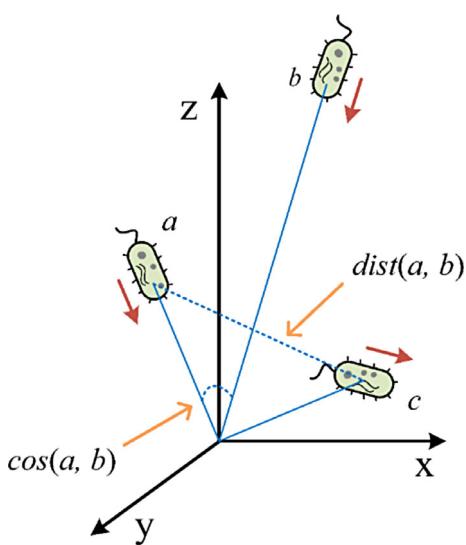


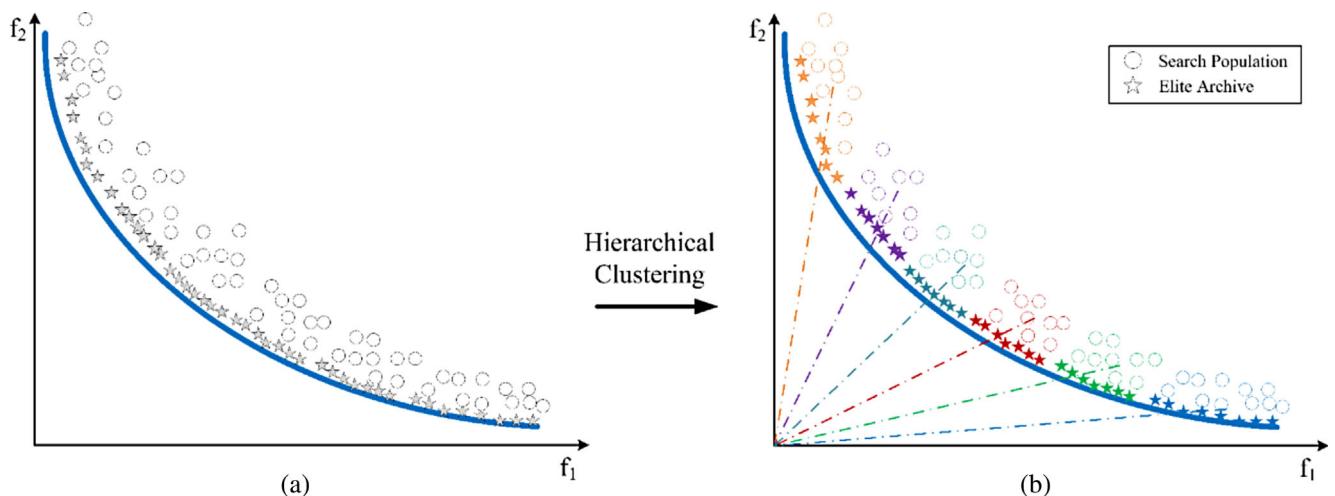
FIGURE 4 Comparison of cosine similarity and Euclidean distance.

Algorithm 1 Dynamic multi-leader selection mechanism.

Input: P (Population), EA (Elite archive), $cluNum$

01: Calculate the best objective value in U ($U = P \cup EA$), z^*
 02: Calculate the worst objective value in U ($U = P \cup EA$), z^-
 03: Normalize the whole population into [0,1] by Eq.(4), $norP$
 04: **for** $i = 1$ to $|U|$ **do**
 05: **for** $j = 2$ to $|U|$
 06: | Calculate cosine distance between individuals in $norP$ by Eq.(5)
 07: **end for**
 08: **end for**
 09: $clusters = HClustering(norP, cosSim, cluNum)$
 /* Run hierarchical clustering based on direction, return cluster label of every individual*/
 10: **for** $p \in P$ **do**
 11: | Calculate the convergence degree of $norP$ by Eq.(6)
 12: **end for**
 13: **for** $q \in EA$ **do**
 14: | Calculate the convergence degree of EA by Eq.(6)
 15: | Calculate the crowding distance in EA by Eq.(7)
 16: **end for**
 17: **for** $c \in clusters$ **do** /* c contains a certain number of searching individuals */
 18: | Find the convergence leader $cleader_c$ in clusters by Eq.(8)
 19: | Find the diversity leader $dleader_c$ in clusters by Eq.(9)
 20: **end for**

Output: $cleader_c, dleader_c$

**FIGURE 5** Direction-based population hierarchical clustering.**2. Leaders selection**

The convergence leaders and diversity leaders are selected to provide the guidance for population evolution. During the optimization process, the minimum distance between the population and the ideal point of a MOP will decrease steadily when the population is continuously evolutionary (Wu et al., 2019). To simulate the leaders with better convergence, the Euclidean distance of the whole population to the ideal point is represented as the convergence degree of individuals which can be calculated by Equation (6).

$$\text{ConvD}_i = \sqrt{\sum_{j=1}^m (p_{ij} - O_j)^2}, \quad (6)$$

where, O represents the normalized ideal point with m dimensions (lines 10–12). Moreover, to obtain some diversity leaders, the crowding distance (Deb, Agrawal, et al., 2002) is applied to represent the diversity degree of every individual. Diversity can be calculated by Equation (7). The larger values of the crowding distance mean the better performance in diversity since the smaller values indicate that the solutions with higher concentration are obtained (lines 13–16).

$$\text{CrowD}_i = \sum_{j=1}^m \left(\frac{f'_j(p_{i+1,j}) - f'_j(p_{i-1,j})}{z_j^- - z_j^+} \right), \quad (7)$$

where, CrowD_i is the crowding distance of the i th bacterium after the whole population being sorted according to each objective function value in ascending order and that the boundary solutions have two times the maximum crowding distance. After that, according to Equation (8), the individual with the minimum distance to the ideal point in the c th bacterial cluster, clusters_c , is defined as a convergence leader for corresponding individuals in clusters_c .

$$\text{cleader}_c = \min_{p_j \in \text{clusters}_c} (\text{ConvD}_j). \quad (8)$$

Additionally, the individual with the maximum crowding distance in the c th bacterial cluster, clusters_c , is selected as a diversity leader for this subpopulation by Equation (9) (lines 17–20).

$$\text{dleader}_c = \max_{p_j \in \text{clusters}_c} (\text{CrowD}_j). \quad (9)$$

Through the multi-leader selection mechanism, cleader and dleader are selected respectively to increase the convergence and diversity of bacterial colony for solving MOPs. Correspondingly, single optimal direction search behaviours are modified to multi-leader directions that represent the balance of convergence and diversity. The modified multi-leader bacterial colony learning behaviours mainly reflect in swimming processes. In addition, to avoid trapping into local Pareto optimal, the elimination behaviour is modified to adaptive change according to convergence status of population instead a fixed frequency.

Swimming and tumbling: The bacterial positions are updated through swimming process using an initial weight ω and a stochastic tumbling operator according Equation (10).

$$p_i = \omega p_i + C_i \cdot \text{Tumble}_c, \quad (10)$$

where, C_i is a random chemotaxis step size for the i th bacterium in each bacterial cluster. The stochastic tumbling operation is realized under guidance of the convergence leader and diversity leader by Equation (11).

$$\text{Tumble}_c = r_{con}(\text{cleader}_c - p_i) + r_{div}(\text{dleader}_c - p_i), \quad (11)$$

where both r_{con} and r_{div} are the learning rates towards convergence direction and diversity direction, respectively.

Elimination: An adaptive elimination is implemented to avoid trapping into the local Pareto optimal. The counter is used to record the times that the bacterial colony cannot search for a better position. The minH records the historical minimum distance to the ideal point. When minH is unchanged or smaller than the current minimum distance to the ideal point minC , the counter will be increased by a unit. Otherwise, it will be reset to zero and the minH is replaced by minC . If the counter is larger than three, the bacteria will be eliminated with a probability Ped which is generated by Equation (12) with a trajectory shown as Figure 6.

$$\text{Ped} = -2\text{counter}^{-1} + 1 \quad \text{counter} \geq 3. \quad (12)$$

From Equation (12) and Figure 6, we can see that the trajectory of elimination probability is an incremental trend, which is equipped with the ability to adjust elimination probability according to counter value.

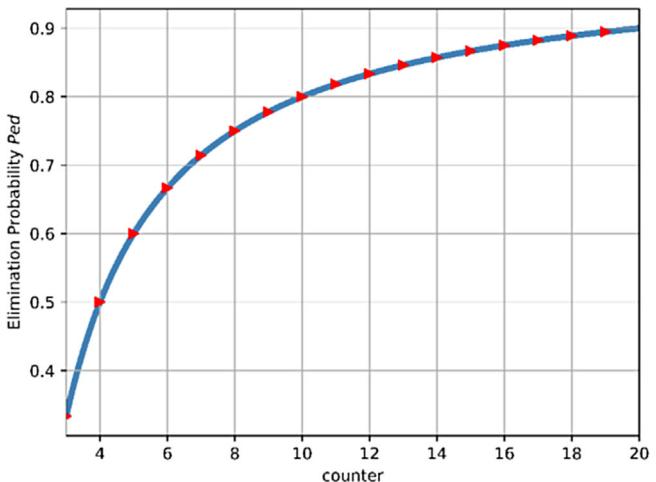


FIGURE 6 The trajectory of elimination probability.

3.2 | Elite archive evolution

It is a characteristic of population-based intelligent optimization algorithms to update the individuals' positions using historic knowledge. The individuals with the relatively optimal positions are kept in elite archive. As a result, there is an overlap between the elite archive and population. This overlap cannot pose a progress in evolution by only learning from elite archive. It is more likely to get another relatively optimal space by adding a randomly searching factor. Therefore, to further improve exploration capability, the elite individuals are designed to have the ability of randomly evolution without learning from others. This coevolution model not only enables the elites to further improve the quality, but also gives more random choices to avoid getting trapped into local optimum.

The elite archive evolution is made up of three parts, including elite duplication for matching, crossover and mutation, elite archive updating. The pseudocode of elite archive evolution is provided in Algorithm 2.

1. Elite duplication for matching.

Considering that the number of the elite archive may be at a low level for effective selection and crossover operations, a control operation of elite duplication is applied and the matching pool MP is set for storing all elites for parent selection. If the number of the elite archive is less than one-eighth of the population, the elite archive will not evolve (lines 1–2). If up to one-eighth of N but less than one-fifth, all individuals in the elite archive will be duplicated to the number of N by Equation (13).

$$duplicatedNum_i = \left\lceil N \frac{CrowD_i}{\sum_{p_j \in EA} CrowD_j} \right\rceil, \quad (13)$$

where, $\lceil \bullet \rceil$ is the ceiling operator. In this case, the crowding distance of EA is calculated by Equation (7) at first (line 5). Next, each elite individual is duplicated to a certain number according to crowding distance and the total number of members in matching pool is expanded to N . The better the diversity, the greater is the value. By Equation (13), individuals with better diversity performance are correspondingly replicated to a greater number by the ratio of crowding distance, which makes it generated more offspring (line 6). All members of the elite are copied to the N with a different number and added to the matching pool MP (line 7). When the number of the elite archive exceeds one-fifth of the population size N , individuals in the elite archive will be selectively copied and added to the matching pool instead of operating on all elite individuals. The crowding distance of all members are calculated and sorted in ascending order, and individuals with higher diversity in one-fifth of the total population N are repeated by Equation (13). Finally, all duplicated individuals are added into the matching pool for parent selection (lines 10–15). Next, all members in matching pool MP are set one parent, which ensures that the elite individuals in the matching pool all have a chance to produce their offspring.

2. Crossover and mutation.

The spouses are selected from MP randomly for subsequence crossover (line 18). The function $SBX(p1, p2)$, the simulated binary crossover, means that two selected parents exchange coding components with a certain rule. In SBX , η is a predefined parameter. The higher the value of η ,

Algorithm 2 Elite archive evolution.

Input: $EA, MP = \emptyset$ (Matching pool), $offspring = \emptyset, N$ (population number)

01: **if** $|EA| \leq N/8$, **then**

02: | Update elite archive $EA = EA$

03: **elseif** $N/8 \leq |EA| \leq N/5$, **then**

04: | **for** $q \in EA$ **do**

05: | Calculate the crowding distance in EA by Eq.(7)

06: | Repeat each individual by Eq.(13)

07: | Add duplicated group into MP

08: | **end for**

09: **else**

10: | **for** $q \in EA$ **do**

11: | Calculate the crowding distance in EA by Eq.(7)

12: | Sort $CrowD$ of each individual with ascending order

13: | Repeat the first $N/5$ individuals by Eq.(13)

14: | Add duplicated group into MP

15: | **end for**

16: **end if**

17: **for** $i = 1$ to N **do**

18: | $p1 = p_i \in MP$, randomly choose a spouse $p2$ from MP

19: | $u = SBX(p1, p2)$

20: | $v = PM(u)$

21: | $offspring = offspring \cup v$

22: **end for**

23: $EA = updatingEA(EA, offspring)$

Output: EA

the higher is the probability that the offspring produced that will be close to the parent. Two groups of offspring are produced after SBX and only half of them μ are chosen randomly into the mutation operator (line 19). To avoid two parents being selected as the same individual and generating invalid offspring, the polynomial mutation PM is performed after SBX, generating a new offspring v that would be added into offspring (lines 20–21). It is noted that not only the algorithms for crossover and mutation are not limited to the two mentioned above, but other forms of evolutionary operators can also be chosen.

3. Elite archive updating.

Finally, the function $updatingEA(EA, offspring)$ indicates that an elite archive EA incorporated $offspring$ is updated according to crowding distance. In the archive updating, all dominating solutions are removed. Besides, the individuals with higher crowding distance, that is, worse diversity, are deleted one by one when the number of the elite archive exceeds maximum capacity. A new elite archive is returned to the next optimization process (line 23).

In this way, the elite archive is evolved based on the genetic idea, which explores more potential space. Different from conventional population-based optimization that the elite ones produced by search population solely, elite archive evolution further improves the diversity of the whole population and avoids trapping into local optimum due to overlap with search population.

3.3 | Multiobjective bacterial colony optimization with dynamic multi-leader co-evolution

The pseudocode of overall MBCO/DML is presented as Algorithm 3. After initialization, the proposed MBCO/DML starts the loop of evolution until it meets the maximum evaluation number (lines 1–29). First, the elite archive evolution strategy is executed with Algorithm 2 and the elite archive is updated (lines 1–2). Then, running a dynamic multi-leader selection mechanism produces convergence leaders and diversity leaders for the next

Algorithm 3 Overview of MBCO/DML.

Initialization: P (Population), EA (Elite archive), $counter = 0$, $minH = \text{Inf}$

- 01: **while** the maximum number of iterations is not satisfied **do**
- 02: Run elite archive evolution with **Algorithm 2**.
- 03: Update elite archive using crowding distance
- 04: Run dynamic multi-leader selection mechanism with **Algorithm 1**.
- 05: **for** $c \in clusters$ **do**
- 06: Update the position of c according to Eq.(10) and Eq.(11)
- 07: $nSwim = 0$ /* Initialize swimming number */
- 08: **while** $nSwim < Ns$ **do**
- 09: **if** any members in c cluster is improved, **then**
- 10: | Update their position according to Eq.(10) and Eq.(11)
- 11: **else**
- 12: | $nSwim = Ns$
- 13: **end if**
- 14: **end while**
- 15: **end for**
- 16: Calculate the convergence degree by Eq.(6)
- 17: Find out the best convergence degree, $minC$
- 18: **if** $minC \geq minH$, **then**
- 19: | $counter = counter + 1$
- 20: **else**
- 21: | $minH = minC$
- 22: | $counter = 0$
- 23: **end if**
- 24: **if** $counter \geq 3$, **then**
- 25: | Eliminate bacteria with Ped produced by Eq.(12)
- 26: **end if**
- 27: Disturbance to duplicated bacteria
- 28: Update the elite archive
- 29: **end while**

Output: EA (Elite archive)

search direction with Algorithm 1 (line 4). After that, the whole population is divided into several clusters with different evolution directions, and the chemotaxis of each bacterium is conducted according to a specific convergence point and a specific diversity one by Equations (10) and (11).

The main characteristic of chemotaxis is to run towards the higher nutrient concentration. Therefore, once the global best of the population is updated, the swimming behaviour is activated to make bacteria walk in the same direction until the performance cannot be further improved or the maximum swimming iteration times Ns is satisfied (lines 5–15). After the chemotaxis process, the main updating operation is completed and a new population is produced. Before the elimination, the convergence degree of the current iteration is calculated by Equation (6), and the current minimum distance to the ideal point is recorded as $minC$ (lines 16–17). Subsequently, the adaptive procedure is executed (lines 18–26). The individuals with the same decision variables are checked and distributed to other decision space points, which is to obtain a group with better diversity (line 27). Finally, the elite archive is updated and the above processes are repeated until the maximum iterations is satisfied (line 28).

4 | EXPERIMENTAL SETUP

4.1 | Benchmarks and experimental designs

In this paper, a series of well-known and challenging test functions are selected to validate the performance of the proposed MBCO/DML. The overview of benchmark problems is provided in Table 1. FON, KUR, POL, and SCH1 2 (Niu, Liu, & Tan, 2019; Yi et al., 2016) are classic low-

TABLE 1 The overview of benchmark problems.

Benchmarks	Variables (D)	Objectives (M)	Features of true PF	Sample size in PF
FON	3	2	Concave	500
KUR	3	2	Disconnected	874
POL	2	2	Convex; disconnected	1102
SCH1	1	2	Convex	500
SCH2	1	2	Disconnected	1335
ZDT1	30	2	Convex	10,000
ZDT2	30	2	Concave	10,000
ZDT3	30	2	Disconnected	10,000
ZDT4	10	2	Convex; multimodal	10,000
ZDT6	10	2	Concave; multimodal; biased	10,000
F1,5	30	2	Convex	10,000
F2,3,6,7,9,10	30	2	Concave	10,000
F4,8	30	3	Concave	9870
DTLZ4	12	3	Concave	9870
DTLZ5,6	12	3	Concave; degenerated; biased	10,000
DTLZ7	22	3	Disconnected; multimodal	10,000

dimension problems, whose features of Pareto front cover almost all distribution types, including concavity, convexity, and disconnectivity. Even so, to verify the validity in high-dimension decision variables problems, the ZDT series functions are applied. To further examine the effectiveness, two other more challenging groups of test functions with tri-objective, that is, F1–10, DTLZ 4–7, are applied. With the previous comparison problems, more local Pareto optimal solutions exist in the F series and DTLZ series to make it more difficult for the multiobjective algorithms. For more relevant details of these benchmark problems, please check (Deb, Thiele, et al., 2002; Zhang et al., 2008), respectively.

We design three group of experiments to demonstrate the performance of the proposed algorithm, that is, ablation experiments of sub-strategies, comparisons with novel multiobjective bacterial-based optimization algorithms (MOBAs), comparisons with other multiobjective evolutionary algorithms (MOEAs).

- *Ablation experiments of sub-strategies:* This experimental design disassembles each proposed sub-strategy and validates their functions by stacking them one by one.
- *Comparisons with MOBAs:* MOBFO (Yi et al., 2016), MORBCO (Niu et al., 2020), and MCBFO (Niu, Liu, & Tan, 2019). These three MOBAs represent the new trend applying bacterial-based structures to solve MOPs in recent years.
- *Comparisons with other population-based multiobjective optimization algorithms:* NMPSO (Lin et al., 2018), BiGE (Li et al., 2015), RVEA (Cheng et al., 2016), and MOCell (Nebro et al., 2009). Those algorithms represent the most dominant idea of population heuristic evolution for intelligence optimization.

4.2 | Settings of parameters

For a fair comparison of the performance differences between the comparison algorithms, the parameters of each algorithm were set according to the reference optimum parameters as reported in Table 2. All experiments are carried out on a PC with AMD 4800H 3200 MHz CPU and 16 GB memory. We carry out our numerical experiments on the platform PlatEMO (Tian et al., 2017), which can be accessed via <https://github.com/BIMK/PlatEMO>.

As the first attempt to integrate clustering concept into the practice of leader selection in population-based multiobjective optimization, the number of clusters $cluNum$ is set to 6 by repeated experiments and getting the best value. As indicated by some population-based algorithms (Lin et al., 2018; Martínez & Coello, 2011; Yi et al., 2016), their initial weight ω is sampled in a range, while keeping the weight and learning rate in its original position. The initial weight ω of MBCO/DML is generated in [0.8, 1.3] and the learning rate for convergence leader r_{con} and diversity leader r_{div} is produced in [1.5, 2.5]. In addition, the chemotaxis step size C is sampled between the lower chemotaxis step size and the upper is set in [0.1, 1.2]. The number of swimming times N_s is set to 2 for MBCO/DML to execute greedy search. For the capability of elite archive is set to 100 as (Yi et al., 2016). For elite archive evolution, p_c and p_m are the crossover probability and the mutation probability respectively that are set

TABLE 2 Parameters setting of all algorithms compared.

Algorithms	Settings of parameters
MOBFO (Yi et al., 2016)	$N_{ed} = 1/N, N_{re} = 1/N, N_c = 1/N, p_{ed} = 1/N, C_i \sim [0.1, 0.2]$ if stagnation
MORBCO (Niu et al., 2020)	$freRc = 10, freRe = 20, N_s = 4, P_{el} = 0.25, C_1, C_2 = 3, C = 0.001, pRep = 1/20, npbest = 2$
MCMBFO (Niu, Liu, & Tan, 2019)	$N_c = 300, N_{re} = N_{ed} = 2, N_s = 4, C_{\max} = 1, C_{\min} = 0.1, C_1 = C_2 = 7.5, C_3 = 10, lInfor_{\max} = 0.002$
NIMPSO (Lin et al., 2018)	$\omega \sim [0.1, 0.5], c_1, c_2, c_3 \sim [1.5, 2.5], p_c = 1/N, \eta_m = 20$
BiGE (Li et al., 2015)	$p_c = 1, \eta_c = \eta_m = 20, p_m = 1/N$
RVEA (Cheng et al., 2016)	$\alpha = 2, fr = 0.1$
MOCell (Nebro et al., 2009)	$p_c = 1, p_m = 1/N, Feedback = 20$
MBCO/DML	$cluNum = 6, \omega \sim [0.8, 1.3], C \sim [0.1, 1.2], r_{con}, r_{div} \sim [1.5, 2.5], N_s = 2, p_c = 0.9, p_m = 1/20, \eta_c = \eta_m = 5$

to 0.9 and 1/20. The larger crossover and mutation probability that is different from previous literature ensures that the crossover and mutation operators can produce more different offspring from the parental generation. Both η_c and η_m being defined as 5 are the distribution factors for SBX and PM respectively.

Among all comparison algorithms, the population size is set to 100 for bi-objective problems and 105 for tri-objective problems. The maximal number of fitness evaluations (FEs) is set at 100,000 for bi-objective problems and at 150,000 for tri-objective problems. All the experimental results are obtained after 30 independent runs. Furthermore, the Wilcoxon rank-sum test (Tian et al., 2018) is employed at the significance level of 0.05. The symbols ‘+’, ‘-’, ‘≈’ provided in the bottom of tables represent that the result is significantly superior to, significantly inferior to and statistically similar to that obtained by MBCO/DML, respectively.

4.3 | Evaluation metrics

For the evaluation metrics, we use three commonly used metrics, that is, inverted generational distance (IGD), hyper-volume (HV) and diversity index *Spread*, to evaluate the quality of the results. Brief descriptions of these metrics are given as follows:

IGD (Bosman & Thierens, 2003) focuses on measuring the average distance between the feasible solutions obtained by the MO algorithms. The better the comprehensive performance and Pareto solutions distribution multiobjective algorithms get, the smaller the IGD value will obtain. It can be calculated using Equation (14).

$$IGD(PS, PF) = \frac{\sum_{x \in PF} \min_{y \in PS} dist(x, y)}{|PF|}, \quad (14)$$

where, PS is obtained for the Pareto solution set and PF represents a uniformly distributed set of reference points sampled from Pareto front, while $dist(x, y)$ is used to calculate the Euclidean distance of x and y .

HV (Beume et al., 2009) is used to measure the volume of the hypercube surrounded by the obtained results and the given reference point. The greater the value of HV, the better is the performance of the multiobjective algorithm. It can be calculated by Equation (15).

$$HV(PS, referp) = \delta \left(\bigcup_{i=1}^{|PS|} v_i \right), \quad (15)$$

where, $referp$ represents a reference point used to calculate the volume of the region in the target space enclosed by the reference point and the set of nondominated solutions obtained by the algorithm. δ denotes the Lebesgue measure, which is used to measure the volume; v_i is the hyper-volume of the reference point and the i th solution in the non-dominated solution set. The reference point used for calculating HV is set to $(1.1, 1.1, \dots, 1.1)^n$.

Spread (Li & Zheng, 2009) is to evaluate the ability of the MO algorithms to achieve better diversity performance, which can be calculated by Equation (16).

$$SP(PS, PF) = \frac{\sum_{i=1}^m mdist(E_i, PS) + \sum_{x \in P} |mdist(x, PS) - \overline{dist}|}{\sum_{i=1}^m mdist(E_i, PS) + (PS| - m) \times \overline{dist}}, \quad (16)$$

where, $E_i(1, 2, \dots, m)$ are the extreme solutions in PF, m is the number of objective function. $mdist(x, PS) = \min_{y \in PS, y \neq x} \|f(x) - f(y)\|$ denotes the minimum gap from the solution x to current solutions set. $\overline{dist} = \sum_{x \in PS} mdist(x, PS) / |PS|$ represents the average minimum distance of solution x with other solutions. Smaller value of *Spread* indicates the better diversity of obtained Pareto front.

5 | EXPERIMENTAL RESULTS AND ANALYSIS

5.1 | Ablation experiments of sub-strategies

In above sections, the main components and strategies of MBCO/DML have been introduced, which includes dynamic multi-leader selection for subpopulations and co-evolution search on elite archive mechanisms. To validate the contribution of each strategy, we compare each key component in this section over tri-objective DTLZ5 benchmark. The visualized results demonstrate the effectiveness of key component of MBCO/DML.

First of all, we extend the standard BCO to solve MOPs with external archive and select an elite as global best randomly. Then, the elite evolution mechanism is added in standard MBCO to verify the improvement that co-evolution brings. On the other hand, the proposed dynamic multi-leader selection mechanism is integrated into the conventional BCO framework but no the assisted elite co-evolution. This uncompleted version is to verify the performance of the proposed dynamic multi-leader evolution mechanism in solving MOPs. Finally, the unabridged MBCO/DML version is conducted to see whether the genetic evolution and the population evolution can reinforce each other by the co-evolution method.

Converging to the true Pareto front is the primary task of multiobjective optimization. In addition, making the solution set with a better diversity based on convergence is another challenge of multiobjective optimization. It can be seen from Figure 7 that the performance of original BCO framework without any modifications is the worst in all ablation experiments. Most bacterial individuals cannot approach the true Pareto front

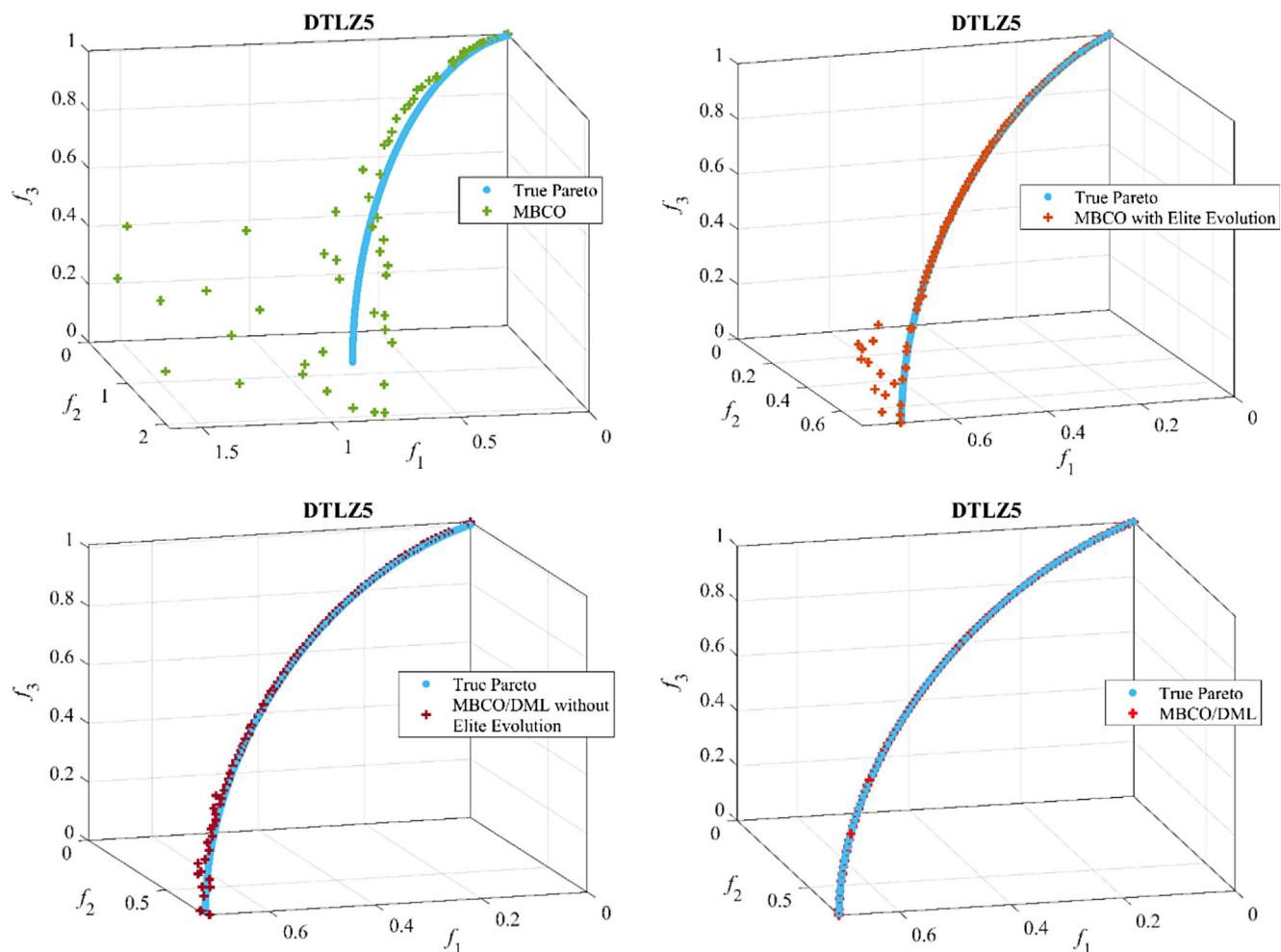


FIGURE 7 The obtained Pareto front on DTLZ5 benchmark.

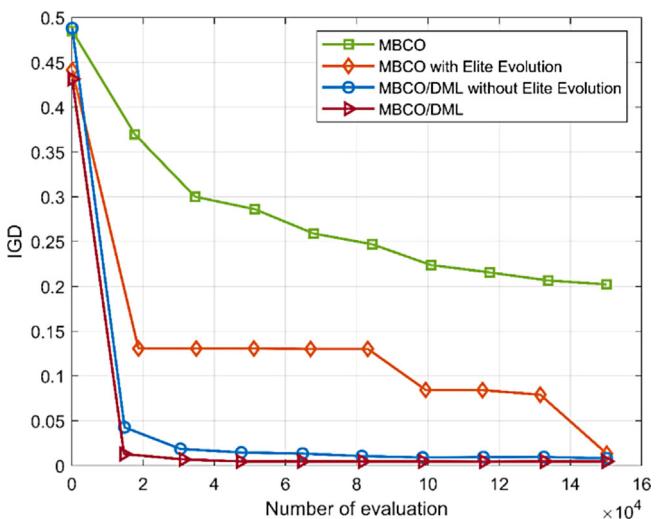


FIGURE 8 The optimization process of IGD on DTLZ5 benchmark.

and a part of the members gather in the local Pareto optimal. The obtained Pareto front by MBCO shows lower convergence and diversity. With the help of elite evolution mechanism, we can see that the most population can approach to true Pareto font but still unable to cover the whole true front. Moreover, due to the single global leader setting, the entire colony cannot cover the true Pareto front totally and lack the key searching capability, especially in some challenging objective space. From the result of MBCO/DML without elite evolution, we can see that almost the whole population can cover the true Pareto front. But due to the bootstrap of elite archive, dynamic multi-leader mechanism still lacks of sufficient convergent capability and diversity maintenance in limited evaluation times.

With assistance of elite co-evolution mechanism, it can be seen that our proposed MBCO/DML can fully cover the true Pareto front and achieve a relative uniform distribution. Figure 8 represents the IGD descent tendency for the standard MBCO and proposed MBCO with different strategies. It can be observed that the IGD value of MBCO/DML declines with the fastest speed and achieve the best performance. Compared to MBCO, our proposed algorithms greatly enhance the capability of BCO to tackle MOPs. In comparison to MBCO with elite evolution, self-tuning elimination probability of MBCO/DML can adaptively adjust the place of bacteria and avoid trapping into local optima. Finally, the elite evolution provides the better leadership for search population and thus speed up the optimization process. Therefore, we can draw a conclusion that all of strategies play an essential role in optimization process, that is, dynamic multi-leader selection greatly improves the convergence and diversity for bacterial colony and the elite co-evolution accelerate the whole process.

5.2 | Comparisons with other MOBAs

The mean and standard deviation of the IGD, HV, Spread values of three novel MOBAs and the proposed MBCO/DML on 24 selected benchmarks are presented in Tables 3–5, respectively. From the overall situation of the three indicators, the proposed MBCO/DML performs significantly better than the three novel MOBAs, such as MOBFO (Yi et al., 2016), MORBCO (Niu et al., 2020), and MCMBFO (Niu, Liu, & Tan, 2019). The best mean of metric for each test instance is highlighted in bold.

As observed in Table 3, the proposed MBCO/DML shows an obvious advantage over the other three competitors on main test instances. To be specific, MBCO/DML gains the best IGD value on 19 out of 24 test instances, while the gains are 0 best IGD values for MOBFO, 5 best IGD values for MORBCO and 0 best IGD value for MCMBFO. More specifically, MORBCO just performs best on F3, F7, F9, F10, F4 and MBCO/DML obtains the best metric over other cases. The proposed MBCO/DML shows significantly superior performance as it achieves better than other MOBAs on the ZDT series, DTLZ series and other chosen benchmarks. Of all 24 benchmarks, the IGD values obtained by MBCO/DML statistically better than MOBFO over all cases, MORBCO over 17 cases, and MCMBFO over 23 cases, respectively.

The HV results of all the 24 test problems are provided in Table 4, from which similar conclusions can be drawn from the IGD results in the HV values. MBCO/DML gains better performance over most benchmarks and achieves 17 best HV results out of 24 test instances. Conversely, the results of paired statistical test of the proposed MBCO/DML with MOBFO, MORBCO and MCMBFO are 1/23/0, 4/20/0, and 2/20/2, respectively. It is noted that the mean and standard deviation take the value of zeros because the obtained Pareto front cannot converge to the reference point for calculating HV metric. Even though MBCO/DML performance well on most MOPs, it still shows a poor ability on the MOPs

TABLE 3 IGD values of MBCO/DML and three novel MOBAs on 24 selected benchmarks.

Problem	MOBFO	MORBCO	MCMBFO	MBCO/DML
FON	2.3826e-1 (7.66e-2) –	8.4436e-3 (9.59e-4) –	7.4968e-3 (1.25e-3) –	4.0413e-3 (1.38e-4)
KUR	3.2773e+0 (6.83e-1) –	1.2292e-1 (2.32e-2) –	5.9768e-2 (1.54e-2) –	3.4919e-2 (6.24e-4)
POL	8.5451e-1 (6.49e-1) –	1.9615e-1 (6.32e-2) –	6.1311e-1 (7.43e-1) –	5.8597e-2 (1.84e-3)
SCH1	2.5813e-1 (8.27e-2) –	3.9121e-2 (4.71e-3) –	1.7565e-2 (1.80e-3) ≈	1.6968e-2 (4.53e-4)
SCH2	2.7936e-1 (9.82e-2) –	5.1494e-2 (7.67e-3) –	4.1929e-2 (5.76e-2) –	2.0160e-2 (2.11e-4)
ZDT1	3.8474e-2 (2.71e-2) –	1.2745e-2 (4.28e-3) –	1.4607e-2 (3.02e-3) –	3.8805e-3 (3.76e-5)
ZDT2	1.2414e-1 (9.53e-2) –	1.2822e-1 (2.39e-1) –	4.8163e-2 (1.31e-1) –	3.9865e-3 (3.85e-5)
ZDT3	6.2901e-2 (3.53e-2) –	3.4189e-2 (1.65e-2) –	1.1566e-2 (2.16e-3) –	4.5420e-3 (5.11e-5)
ZDT4	7.7114e+1 (1.13e+1) –	4.8013e-2 (9.62e-2) –	2.7739e+1 (1.41e+1) –	3.8773e-3 (9.94e-5)
ZDT6	1.4072e-2 (8.97e-3) –	1.1294e-2 (5.36e-3) –	7.0528e-2 (2.25e-1) –	3.5153e-3 (2.96e-4)
F1	3.5022e-1 (3.01e-2) –	8.4938e-3 (9.87e-4) –	3.9440e-1 (7.79e-2) –	4.3606e-3 (1.06e-4)
F2	6.0949e-1 (1.13e-16) –	9.1837e-3 (1.05e-3) –	6.0852e-1 (1.59e-3) –	4.7216e-3 (1.47e-4)
F3	4.3130e-1 (4.41e-2) –	1.0962e-2 (2.44e-3) +	2.1523e+0 (1.33e-1) –	3.1900e-1 (2.56e-2)
F5	5.4728e-1 (4.03e-2) –	1.5618e-2 (3.90e-3) –	5.5194e-1 (1.17e-2) –	7.4515e-3 (4.76e-4)
F6	4.2197e-1 (9.61e-3) –	1.3285e-2 (1.77e-3) –	4.0803e-1 (6.26e-3) –	8.7294e-3 (7.45e-4)
F7	4.1578e-1 (3.71e-2) –	2.7338e-1 (8.17e-2) +	1.6091e+0 (1.91e-1) –	3.6300e-1 (3.23e-3)
F9	2.0982e-1 (1.74e-2) –	1.1911e-2 (1.13e-3) ≈	1.7049e-1 (1.07e-2) –	1.7410e-2 (1.00e-2)
F10	8.2316e-1 (7.27e-3) –	1.5505e-1 (2.03e-1) +	8.6108e-1 (2.04e-1) –	7.3785e-1 (9.94e-3)
F4	5.4143e-1 (9.42e-4) –	7.3228e-2 (6.13e-3) +	1.1361e+0 (2.23e-1) –	7.8808e-2 (3.67e-3)
F8	5.4065e-1 (1.19e-3) –	3.0413e-1 (9.16e-2) ≈	4.9981e-1 (5.86e-2) –	2.4890e-1 (1.22e-1)
DTLZ4	9.5398e-1 (1.12e-1) –	5.6856e-1 (1.01e-1) –	1.6662e-1 (1.40e-2) –	7.7999e-2 (5.45e-3)
DTLZ5	4.1180e-1 (5.17e-2) –	1.2453e-1 (6.29e-2) –	3.0495e-1 (5.16e-3) –	4.6877e-3 (1.51e-4)
DTLZ6	1.2341e-2 (1.24e-2) –	5.8239e-1 (5.90e-1) –	3.1912e-1 (1.01e-1) –	4.5389e-3 (6.14e-5)
DTLZ7	5.3066e-1 (1.12e-1) –	1.0077e-1 (1.40e-2) ≈	1.1640e-1 (1.42e-2) –	9.7347e-2 (7.99e-3)
+/-/≈	0/24/0	4/17/3	0/23/1	/
Best/All	0/24	5/24	0/24	19/24

with many local Pareto fronts such as F3, F7, and F10. The convergence pressure can be seen from Figure 9, from which it can be observed that the proposed MBCO/DML still cannot cover all the true Pareto front in the limited evaluation times.

Based on the statistical results of *Spread* from Table 5, the proposed MBCO/DML can achieve a more uniform Pareto front and gain better diversity measurement than the three new MOBAs. With the help of the clustering-based dynamic multi-leader selection mechanism, MBCO/DML obtains the best *Spread* values on 20 out of 24 test instances while other MOBAs such as MOBFO with 1 best *Spread* result on F7 and MCMBFO with 3 best *Spread* values on ZDT6, DTLZ4, and DTLZ7, respectively. MCMBFO introduces a multi-swarm cooperative mechanism for exploring more unknown areas which can help MCMBFO cover a wider range of true Pareto front and enhance the diversity of the final obtained Pareto front. In summary, we can observe that the MBCO/DML shows a great improvement in convergence and diversity. Three novel MOBAs all apply different strategies expanding the original BFO process to solve MOPs. However, since the three-level nested loop search structure is retained, more computing resources and computation time is still required for them while the MBCO/DML can save computation resource by using a parallel structure.

5.3 | Comparisons with other population-based multiobjective optimization algorithms

The mean and standard deviation of the IGD, HV, *Spread* values of the selected MOEAs and the proposed MBCO/DML on 24 selected benchmarks are presented in Tables 6–8, respectively. As seen from the three tables, the proposed MBCO/DML performs significantly better than other commonly effective MOEAs, such as NMPSO (Lin et al., 2018), BiGE (Li et al., 2015), RVEA (Cheng et al., 2016), and MOCell (Nebro et al., 2009). The best mean of metric for each test is highlighted in bold.

TABLE 4 HV values of MBCO/DML and three novel MOBAs on 24 selected benchmarks.

Problem	MOBFO	MORBCO	MCMBFO	MBCO/DML
FON	3.2092e-1 (7.00e-2) -	6.3504e-1 (1.36e-3) -	6.3077e-1 (3.08e-3) -	6.4073e-1 (1.11e-4)
KUR	3.3698e-1 (2.37e-1) -	7.0380e-1 (2.73e-3) -	7.4248e-1 (1.28e-2) +	7.1474e-1 (1.60e-4)
POL	1.0743e+0 (6.90e-3) -	1.0786e+0 (3.74e-3) -	1.0779e+0 (3.74e-3) -	1.0830e+0 (2.23e-5)
SCH1	1.0157e+0 (2.32e-2) -	1.0653e+0 (9.60e-4) -	1.0692e+0 (2.43e-4) ≈	1.0693e+0 (4.16e-5)
SCH2	8.5935e-1 (2.04e-2) -	8.7901e-1 (2.23e-3) -	8.8151e-1 (4.15e-3) +	8.8105e-1 (2.46e-5)
ZDT1	8.8902e-1 (2.60e-2) -	9.1830e-1 (4.19e-3) -	9.1220e-1 (4.46e-3) -	9.3052e-1 (4.14e-5)
ZDT2	4.7601e-1 (9.71e-2) -	5.5639e-1 (1.72e-1) -	6.0669e-1 (9.47e-2) -	6.5511e-1 (4.20e-5)
ZDT3	8.9310e-1 (6.08e-2) +	8.0814e-1 (3.10e-2) -	8.0546e-1 (1.79e-3) -	8.0985e-1 (1.91e-5)
ZDT4	0.0000e+0 (0.00e+0) -	8.6431e-1 (1.37e-1) -	0.0000e+0 (0.00e+0) -	9.3035e-1 (2.72e-4)
ZDT6	5.6292e-1 (7.88e-3) -	5.5343e-1 (1.34e-2) -	5.4140e-1 (9.18e-2) ≈	5.7290e-1 (3.17e-4)
F1	5.3800e-1 (2.38e-2) -	9.2553e-1 (8.52e-4) -	5.3494e-1 (3.21e-2) -	9.2905e-1 (1.95e-4)
F2	2.1000e-1 (1.41e-16) -	6.4985e-1 (9.52e-4) -	2.0991e-1 (5.98e-5) -	6.5306e-1 (2.82e-4)
F3	2.1000e-1 (5.53e-10) -	5.6509e-1 (3.43e-3) +	0.0000e+0 (0.00e+0) -	2.3854e-1 (1.64e-2)
F5	4.6741e-1 (1.08e-2) -	9.1479e-1 (5.38e-3) -	4.7609e-1 (2.87e-3) -	9.2418e-1 (6.33e-4)
F6	2.3640e-1 (1.06e-3) -	6.4218e-1 (3.24e-3) -	2.4233e-1 (1.18e-3) -	6.4599e-1 (1.14e-3)
F7	2.1000e-1 (7.04e-10) -	2.7776e-1 (5.83e-2) +	0.0000e+0 (0.00e+0) -	2.1019e-1 (7.32e-4)
F9	6.8794e-1 (8.85e-3) -	9.1830e-1 (1.78e-3) +	7.2100e-1 (1.02e-2) -	9.0843e-1 (1.45e-2)
F10	2.2991e-1 (8.58e-3) -	7.9401e-1 (1.55e-1) +	2.2794e-1 (5.52e-2) -	3.2658e-1 (1.04e-2)
F4	6.0242e-1 (8.52e-3) -	8.7282e-1 (5.76e-3) +	0.0000e+0 (0.00e+0) -	8.4616e-1 (6.07e-3)
F8	6.0852e-1 (5.16e-3) -	7.4309e-1 (3.34e-2) -	5.9859e-1 (6.64e-2) -	7.6740e-1 (3.53e-2)
DTLZ4	6.3052e-3 (2.74e-2) -	3.1103e-1 (9.78e-2) -	6.8930e-1 (2.74e-2) -	8.5600e-1 (5.87e-3)
DTLZ5	1.1295e-2 (1.36e-2) -	2.3256e-1 (8.47e-2) -	8.2924e-2 (3.54e-3) -	4.0051e-1 (8.11e-5)
DTLZ6	3.7440e-1 (3.20e-2) -	1.9137e-1 (1.58e-1) -	9.5088e-2 (2.03e-2) -	4.0075e-1 (3.41e-5)
DTLZ7	3.9194e-1 (2.21e-2) -	4.6882e-1 (7.07e-3) -	4.4348e-1 (8.46e-3) -	4.7680e-1 (4.61e-3)
+/-/≈	1/23/0	4/20/0	2/20/2	/
Best/All	1/24	4/24	2/24	17/24

Table 6 shows the statistical results of IGD values of the proposed MBCO/DML and four MOEAs for 24 test instances. It can be observed that the proposed MBCO/DML obtains the 18 best IGD values out of 24 benchmarks, which is much greater than the other four competitive MOEAs. While there are 0 best IGD results for BiGE, MOCell and only 2 best IGD values for RVEA on F8, DTLZ4, 4 best for NMPSO on F3, F4, F7, and DTLZ7. In contrast, the proposed MBCO/DML can achieve superior performance over most test instances that reach an accuracy level under 10e-3. As for paired statistic test results, the MBCO/DML significantly outperforms NMPSO over 19 cases, BiGE over 23 cases, RVEA over 22 cases, MOCell over 23 cases, respectively. The proposed MBCO/DML exhibits a significant advantage over the other MOEAs on bi-objective problems except NMPSO on F3, F7. In terms of tri-objective optimization problems, that is, F4, F8, DTLZ4–DTLZ7, the proposed MBCO/DML showed a significant advantage, especially in degenerated and biased DTLZ5 and DTLZ6. In generally, the comprehensive IGD metric results show MBCO/DML have a significant competitive edge on bi- and tri-objective optimization problems.

As seen in Table 7, the 20 best HV values are obtained by the proposed MBCO/DML, while 1 best HV value for NMPSO and MOCell respectively, 2 best HV values for RVEA are obtained. For paired test on HV metric results, the values of MBCO/DML are significantly better than NMPSO over 22 cases, BiGE over 23 cases, RVEA over 22 cases, and MOCell over 23 cases, respectively. It is seen that the proposed MBCO/DML outperforms almost all algorithms on bi-objective benchmarks except the MOCell on ZDT3. For tri-objective benchmarks of the F series and DTLZ series, only the RVEA performs better on F8 and DTLZ4 and NMPSO on DTLZ7 than MBCO/DML.

Table 8 shows the diversity measurement *Spread* of obtained Pareto front, which is shown to demonstrate that our proposed clustering-based dynamic multi-leaders selection mechanism can efficiently realize better diversity, that is, the evenness of the final Pareto front. From the statistical results of *Spread* values, 17 best performances are obtained by the proposed MBCO/DML, while 0 best *Spread* value is obtained for BiGE, NMPSO, and MOCell, respectively. It is noticed that RVEA applying decomposition concept could reach 7 best performances on overall 24 test instances. RVEA still shows relatively strong competitiveness in enhancing diversity for its adaptive adjustment reference vector-guided mechanism. The diversity leader selection mechanism used by MBCO/DML achieves better performances in enhancing the diversity of obtained

TABLE 5 Spread values of MBCO/DML and three novel MOBAs on 24 selected benchmarks.

Problem	MOBFO	MORBCO	MCMBFO	MBCO/DML
FON	9.3634e-1 (2.45e-1) -	1.0538e+0 (9.37e-2) -	2.2344e-1 (1.93e-2) -	1.5492e-1 (1.26e-2)
KUR	1.0136e+0 (2.27e-1) -	1.3594e+0 (6.84e-2) -	4.7291e-1 (7.09e-2) -	2.0532e-1 (2.11e-2)
POL	1.0956e+0 (2.80e-1) -	1.3110e+0 (8.45e-2) -	7.2727e-1 (2.77e-1) -	1.8655e-1 (2.01e-2)
SCH1	9.3160e-1 (2.00e-1) -	1.0692e+0 (8.57e-2) -	2.0708e-1 (2.64e-2) -	1.7537e-1 (1.53e-2)
SCH2	1.0131e+0 (1.89e-1) -	1.1740e+0 (8.98e-2) -	3.8276e-1 (1.80e-1) +	4.5059e-1 (2.39e-2)
ZDT1	6.6209e-1 (2.64e-1) -	1.3301e+0 (1.81e-1) -	2.2303e-1 (3.09e-2) -	1.7281e-1 (1.62e-2)
ZDT2	1.0711e+0 (1.51e-1) -	1.4118e+0 (3.38e-1) +	3.2844e-1 (1.67e-1) -	1.6079e-1 (1.80e-2)
ZDT3	1.2189e+0 (1.29e-1) -	1.5807e+0 (2.09e-1) -	2.8625e-1 (3.53e-2) -	2.1009e-1 (2.24e-2)
ZDT4	1.0097e+0 (4.89e-2) -	1.3033e+0 (1.25e-1) -	1.0019e+0 (2.40e-2) -	1.3962e-1 (1.49e-2)
ZDT6	1.3870e+0 (4.67e-1) -	1.7759e+0 (1.49e-1) -	7.5600e-1 (4.37e-1) ≈	7.9691e-1 (4.69e-1)
F1	1.1517e+0 (6.58e-2) -	9.7414e-1 (7.71e-2) -	1.0132e+0 (1.24e-1) -	1.6673e-1 (1.25e-2)
F2	1.0000e+0 (0.00e+0) -	9.8729e-1 (9.08e-2) -	1.0000e+0 (8.52e-4) -	1.6901e-1 (1.09e-2)
F3	1.1358e+0 (1.76e-1) -	1.1242e+0 (1.22e-1) -	9.3745e-1 (2.85e-2) -	8.7042e-1 (1.06e-1)
F5	1.0023e+0 (4.47e-2) -	1.2784e+0 (1.13e-1) -	8.6152e-1 (9.78e-3) -	1.6015e-1 (1.32e-2)
F6	1.0282e+0 (1.42e-2) -	1.2495e+0 (7.23e-2) -	9.1809e-1 (9.16e-3) -	1.7321e-1 (1.51e-2)
F7	9.0514e-1 (1.14e-1) +	1.3728e+0 (1.13e-1) -	8.9084e-1 (2.61e-2) +	9.5294e-1 (1.06e-1)
F9	9.0370e-1 (7.32e-2) -	1.2789e+0 (9.08e-2) -	7.7150e-1 (1.02e-1) -	1.8046e-1 (2.47e-2)
F10	1.0114e+0 (6.53e-3) -	1.3178e+0 (2.02e-1) -	9.9595e-1 (1.63e-3) -	9.5858e-1 (5.52e-3)
F4	1.6272e+0 (1.10e-1) -	8.0502e-1 (1.73e-1) -	6.9994e-1 (1.77e-1) -	5.8167e-1 (2.24e-1)
F8	1.6337e+0 (1.68e-1) -	1.5254e+0 (2.37e-1) -	1.1845e+0 (2.46e-1) -	4.9563e-1 (2.51e-1)
DTLZ4	1.3069e+0 (1.63e-1) -	1.2362e+0 (9.55e-2) -	4.8899e-1 (8.32e-2) +	5.5522e-1 (5.99e-2)
DTLZ5	6.4469e-1 (5.63e-2) -	1.3083e+0 (1.36e-1) -	6.3678e-1 (5.07e-2) -	2.4243e-1 (2.11e-2)
DTLZ6	1.0409e+0 (5.51e-1) -	1.4784e+0 (4.56e-1) -	8.4543e-1 (2.11e-1) -	2.3355e-1 (1.63e-2)
DTLZ7	1.1642e+0 (1.04e-1) -	1.2902e+0 (1.17e-1) -	4.5037e-1 (2.90e-2) +	5.1079e-1 (4.28e-2)
+/-/≈	1/23/0	1/23/0	4/19/1	/
Best/All	0/24	0/24	4/24	20/24

Pareto front. This means that designing different strategies for different spaces can effectively take into account multiple search spaces and obtain good diversity performance. In the paired statistical test, MBCO/DML performances significantly better than NMPSO over 21 cases, BiGE over 23 cases, RVEA over 15 cases, and NMPSO over 21 cases. To further demonstrate the performance of the proposed MBCO/DML, the Pareto front obtained by MBCO/DML is shown in Figure 9, from which we can observe that almost all non-dominated solutions set can achieve good distribution. Overall, the results show that the proposed MBCO/DML is effective and efficient for convergence and diversity with the most MOPs.

5.4 | Effectiveness analysis of MBCO/DML

1. The robustness of dynamic multi-leader strategy

Motivated by the idea of the searching population should have the ability to explore different solution spaces with different evolutionary directions, we dynamically split the entire population into several subgroups with hierarchical clustering. The dynamic multi-leader strategy overcomes the lower robustness of conventional population-based algorithms, that is, easier trapping into local Pareto optimal and lower diversity, caused by solely global guidance mechanisms. In MBCO/DML, dynamic subpopulations have own evolution directions and leaderships. This parallel idea greatly improves the searching efficiency and makes the most individuals follow the best leaderships. From the statistical results with NMPSO, MOBFO, MORBCO, MCMBFO whose searching population is just guided by a single global leader like the best diversity leader, the MBCO/DML is much more capable of searching unknown regions and achieving the performance of better balance of convergence and diversity.

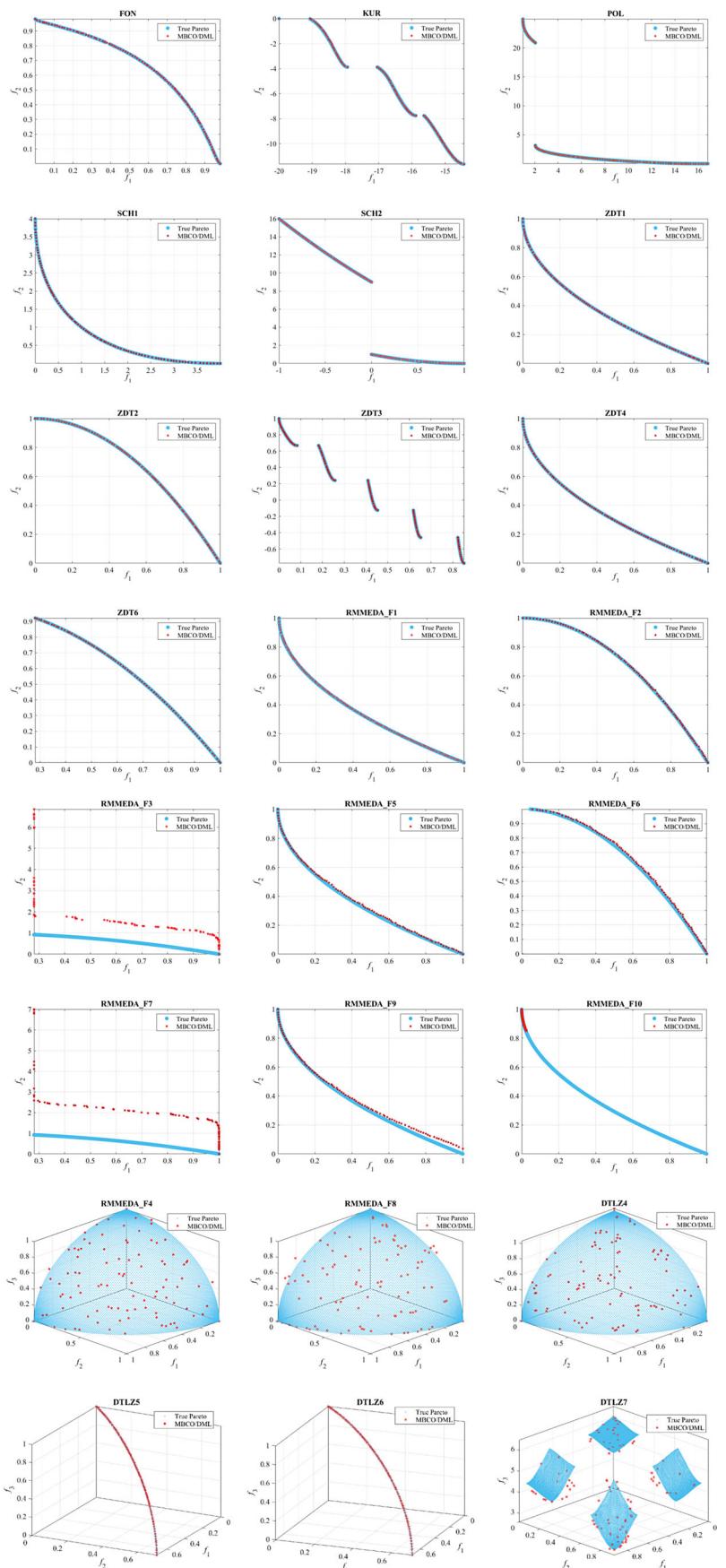


FIGURE 9 The Pareto front obtained by the proposed MBCO/DML.

TABLE 6 IGD values of MBCO/DML and four competitive MOEAs on 24 selected benchmarks.

Problem	NMPSO	BiGE	RVEA	MOCell	MBCO/DML
FON	4.6594e-3 (2.99e-4) –	1.5195e-2 (1.39e-2) –	4.5398e-3 (6.59e-4) –	5.2676e-3 (1.87e-4) –	4.0413e-3 (1.38e-4)
KUR	8.1091e-1 (2.07e-1) –	1.3363e-1 (1.09e-1) –	7.6660e-2 (2.10e-2) –	4.1884e-2 (2.35e-3) –	3.4919e-2 (6.24e-4)
POL	9.4282e+0 (2.95e+0) –	3.4988e+0 (4.69e+0) –	2.7037e-1 (1.39e-2) –	6.8636e-2 (3.38e-3) –	5.8597e-2 (1.84e-3)
SCH1	2.5371e-1 (6.58e-2) –	4.0904e-2 (4.08e-3) –	4.2742e-2 (2.61e-4) –	1.9704e-2 (8.99e-4) –	1.6968e-2 (4.53e-4)
SCH2	1.2882e-1 (4.22e-2) –	5.8770e-2 (8.12e-3) –	4.4828e-2 (1.83e-4) –	2.3691e-2 (7.77e-4) –	2.0160e-2 (2.11e-4)
ZDT1	2.7494e-2 (9.59e-3) –	1.2250e-2 (3.31e-3) –	5.6097e-3 (6.30e-4) –	4.9267e-3 (1.60e-4) –	3.8805e-3 (3.76e-5)
ZDT2	1.8697e-2 (3.61e-3) –	1.2987e-2 (3.51e-3) –	5.6429e-3 (1.42e-3) –	4.9060e-3 (1.72e-4) –	3.9865e-3 (3.85e-5)
ZDT3	1.0114e-1 (6.55e-4) –	1.3092e-2 (1.79e-3) –	8.3414e-3 (5.30e-4) –	6.3463e-3 (5.31e-3) –	4.5420e-3 (5.11e-5)
ZDT4	3.1952e-2 (2.52e-2) –	1.5102e-2 (4.99e-3) –	5.5103e-2 (2.99e-2) –	8.3517e-3 (1.57e-3) –	3.8773e-3 (9.94e-5)
ZDT6	4.4264e-3 (3.64e-4) –	8.3531e-3 (9.39e-4) –	4.4365e-2 (7.10e-3) –	4.9140e-3 (3.35e-4) –	3.5153e-3 (2.96e-4)
F1	1.7389e-2 (5.61e-3) –	2.5972e-1 (3.95e-2) –	1.7320e-1 (3.95e-2) –	1.5763e-1 (4.73e-2) –	4.3606e-3 (1.06e-4)
F2	4.1269e-1 (2.83e-1) –	5.5642e-1 (5.40e-2) –	3.3287e-1 (1.11e-1) –	3.7738e-1 (1.33e-1) –	4.7216e-3 (1.47e-4)
F3	1.5732e-1 (1.40e-2) +	5.9861e-1 (8.22e-3) –	7.3586e-1 (4.51e-2) –	6.0238e-1 (1.21e-2) –	3.1900e-1 (2.56e-2)
F5	2.1436e-2 (6.69e-3) –	3.8858e-1 (1.39e-2) –	3.4123e-1 (4.19e-2) –	3.2938e-1 (2.95e-2) –	7.4515e-3 (4.76e-4)
F6	2.0061e-2 (4.19e-3) –	2.1877e-1 (9.35e-3) –	2.9777e-1 (1.59e-2) –	2.8056e-1 (2.48e-2) –	8.7294e-3 (7.45e-4)
F7	2.6523e-1 (1.27e-1) +	5.8247e-1 (6.47e-3) –	6.8918e-1 (3.54e-2) –	5.8262e-1 (6.22e-3) –	3.6300e-1 (3.23e-3)
F9	4.0520e-2 (2.13e-2) –	6.5690e-2 (3.28e-2) –	5.6837e-2 (1.23e-2) –	2.2644e-2 (8.32e-3) –	1.7410e-2 (1.00e-2)
F10	7.4608e-1 (1.81e-2) ≈	8.0928e-1 (1.57e-3) –	4.8698e+0 (2.08e+1) –	8.0909e-1 (1.93e-3) –	7.3785e-1 (9.94e-3)
F4	7.6668e-2 (3.00e-3) ≈	2.9719e-1 (7.42e-2) –	1.1956e-1 (1.82e-2) –	2.8339e-1 (3.45e-2) –	7.8808e-2 (3.67e-3)
F8	3.3907e-1 (7.09e-2) –	3.7835e-1 (1.65e-2) –	1.2152e-1 (1.67e-2) +	3.0729e-1 (2.39e-2) ≈	2.4890e-1 (1.22e-1)
DTLZ4	9.0024e-2 (8.55e-2) –	1.0493e-1 (1.59e-1) ≈	6.6673e-2 (8.97e-2) +	1.8563e-1 (3.03e-1) –	7.7999e-2 (5.45e-3)
DTLZ5	1.4996e-2 (3.16e-3) –	1.5251e-2 (3.44e-3) –	5.8237e-2 (6.84e-4) –	5.8974e-3 (3.40e-4) –	4.6877e-3 (1.51e-4)
DTLZ6	1.4085e-2 (2.80e-3) –	7.0505e-1 (4.11e-2) –	6.4077e-2 (1.18e-2) –	5.1739e-3 (2.32e-4) –	4.5389e-3 (6.14e-5)
DTLZ7	6.6254e-2 (3.11e-3) +	2.8714e-1 (4.45e-1) –	1.0428e-1 (4.10e-3) –	1.1435e-1 (9.31e-2) –	9.7347e-2 (7.99e-3)
+/-/≈	3/19/2	0/23/1	2/22/0	0/23/1	/
Best/All	4/24	0/24	2/24	0/24	18/24

2. The balance of convergence and diversity

In MBCO/DML, the convergence and diversity leaders are defined to guide the whole population towards a balance evolutionary direction. The Pareto front obtained by the proposed MBCO/DML is represented as Figure 9, from which we can see that MBCO/DML on almost all benchmarks can converge to the true Pareto front and obtain a set of solutions with uniform distribution. From the results of a comparison experiment with NMPSO, MOBFO, MORBCO, MCMBFO that solely depend on a single global optimum to guide the whole population, the proposed MBCO/DML will be able to make full use of the two types of prior knowledge of convergence and diversity simultaneously. On the one hand, two groups of comparison experiments with MOBAs and MOEAs demonstrate our convergence leadership and adaptive elite evolution that can further help the entire population approach to true Pareto front. On the other hand, as we can observe from the tables of Spread, the MBCO/DML completely overwhelms other comparison algorithms in terms of diversity, taking 20/24, 17/24 lead with MOBAs and MOEAs in Spread metric respectively. We can see that the MBCO/DML can obtain a relatively even Pareto front with the guidance of diversity leadership.

3. The performance of elite co-evolution technique

Elite co-evolution technique is used to provide more diverse learning options for jumping out local optimal and speeding up convergence. As shown by the statistical results, NMPSO and MOBAs using the external archive without evolutionary capability are difficult to escape the trap of local Pareto optimal and always get the worst performance. This even cannot converge to reference point for calculating HV metric (e.g., MOBFO on F4, MCMBFO on ZDT4, F3, F7, F4, and RVEA on F3, F7). It is a new operator that incorporates genetic evolution into the population generation. We can observe that MBCO/DML outperforms chosen MOEAs on almost all test instances, which demonstrates that the genetic evolution

TABLE 7 HV values of MBCO/DML and four competitive MOEAs on 24 selected benchmarks.

Problem	NMPSO	BiGE	RVEA	MOCell	MBCO/DML
FON	6.4017e-1 (1.38e-4) –	6.2317e-1 (2.25e-2) –	6.3908e-1 (1.24e-3) –	6.3861e-1 (2.32e-4) –	6.4073e-1 (1.11e-4)
KUR	6.7701e-1 (9.17e-3) –	7.1285e-1 (1.34e-2) –	7.0724e-1 (3.31e-3) –	7.1389e-1 (2.04e-4) –	7.1474e-1 (1.60e-4)
POL	1.0307e+0 (3.04e-2) –	1.0734e+0 (6.07e-3) –	1.0810e+0 (1.36e-4) –	1.0827e+0 (6.86e-5) –	1.0830e+0 (2.23e-5)
SCH1	1.0329e+0 (1.28e-2) –	1.0651e+0 (9.36e-4) –	1.0681e+0 (1.12e-5) –	1.0687e+0 (1.50e-4) –	1.0693e+0 (4.16e-5)
SCH2	8.7505e-1 (4.35e-3) –	8.7878e-1 (1.77e-3) –	8.7804e-1 (8.86e-5) –	8.8073e-1 (8.65e-5) –	8.8105e-1 (2.46e-5)
ZDT1	6.9151e-1 (1.20e-2) –	9.1950e-1 (5.08e-3) –	9.2676e-1 (9.94e-4) –	9.2858e-1 (2.46e-4) –	9.3052e-1 (4.14e-5)
ZDT2	4.3567e-1 (2.12e-3) –	6.4563e-1 (3.52e-3) –	6.5191e-1 (1.77e-3) –	6.5357e-1 (2.07e-4) –	6.5511e-1 (4.20e-5)
ZDT3	5.6681e-1 (2.98e-4) –	8.0585e-1 (7.77e-4) –	8.0485e-1 (1.04e-3) –	8.1214e-1 (1.62e-2) +	8.0985e-1 (1.91e-5)
ZDT4	6.8663e-1 (2.35e-2) –	9.1162e-1 (7.02e-3) –	8.5637e-1 (3.62e-2) –	9.2147e-1 (2.64e-3) –	9.3035e-1 (2.72e-4)
ZDT6	3.8777e-1 (3.12e-4) –	5.6776e-1 (9.72e-4) –	5.1087e-1 (1.02e-2) –	5.6871e-1 (6.75e-4) –	5.7290e-1 (3.17e-4)
F1	7.0192e-1 (7.10e-3) –	7.2388e-1 (2.52e-2) –	7.7381e-1 (2.17e-2) –	7.9596e-1 (3.16e-2) –	9.2905e-1 (1.95e-4)
F2	2.0576e-1 (1.65e-1) –	2.1633e-1 (1.11e-2) –	3.0707e-1 (6.79e-2) –	2.8864e-1 (7.71e-2) –	6.5306e-1 (2.82e-4)
F3	2.2961e-1 (1.15e-2) –	1.7606e-2 (3.58e-2) –	0.0000e+0 (0.00e+0) –	1.1951e-2 (3.10e-2) –	2.3854e-1 (1.64e-2)
F5	6.9612e-1 (8.62e-3) –	6.3230e-1 (8.44e-3) –	6.6574e-1 (2.85e-2) –	6.7576e-1 (1.92e-2) –	9.2418e-1 (6.33e-4)
F6	4.3458e-1 (2.45e-3) –	3.7811e-1 (6.71e-3) –	3.1544e-1 (9.45e-3) –	3.2861e-1 (1.63e-2) –	6.4599e-1 (1.14e-3)
F7	1.6032e-1 (5.88e-2) –	8.7186e-2 (2.44e-2) –	0.0000e+0 (0.00e+0) –	8.1764e-2 (3.29e-2) –	2.1019e-1 (7.32e-4)
F9	6.7193e-1 (2.45e-2) –	8.5685e-1 (2.53e-2) –	8.5043e-1 (1.65e-2) –	8.9657e-1 (1.14e-2) –	9.0843e-1 (1.45e-2)
F10	1.8714e-1 (1.70e-2) –	2.4668e-1 (1.79e-3) –	2.2073e-1 (7.49e-2) –	2.4684e-1 (2.19e-3) –	3.2658e-1 (1.04e-2)
F4	5.6220e-1 (1.13e-3) –	7.2203e-1 (5.10e-2) –	7.7934e-1 (1.77e-2) –	7.2953e-1 (1.13e-2) –	8.4616e-1 (6.07e-3)
F8	4.1683e-1 (3.86e-2) –	6.6858e-1 (3.01e-2) –	7.8489e-1 (2.04e-2) ≈	7.1729e-1 (2.29e-2) –	7.6740e-1 (3.53e-2)
DTLZ4	8.8439e-1 (5.25e-2) +	8.5177e-1 (1.17e-1) –	8.8445e-1 (5.24e-2) +	7.7634e-1 (2.18e-1) –	8.5600e-1 (5.87e-3)
DTLZ5	3.9531e-1 (1.17e-3) –	3.9151e-1 (2.17e-3) –	3.5400e-1 (1.21e-3) –	3.9851e-1 (2.82e-4) –	4.0051e-1 (8.11e-5)
DTLZ6	3.9592e-1 (1.06e-3) –	2.3293e-1 (3.87e-3) –	3.4221e-1 (1.25e-2) –	4.0020e-1 (1.22e-4) –	4.0075e-1 (3.41e-5)
DTLZ7	4.8886e-1 (1.99e-3) +	4.4574e-1 (7.71e-2) ≈	4.7258e-1 (3.54e-3) –	4.6903e-1 (1.65e-2) –	4.7680e-1 (4.61e-3)
+/-/≈	2/22/0	0/23/1	1/22/1	1/23/0	/
Best/All	1/24	0/24	2/24	1/24	20/24

and the population evolution can reinforce each other by the co-evolution method. With the help of the proposed adaptive leader learning mechanism, MBCO/DML can converge to a final Pareto front that almost cover all true Pareto front with uniform distribution for bi-objective optimization problems.

6 | CONCLUSION AND FUTURE WORK

In this article, we propose a new multiobjective bacterial colony optimization with dynamic multi-leader co-evolution, abbreviated as MBCO/DML, for balancing the diversity and convergence in solving MOPs. The dynamic multi-leader concept is proposed to assist with the identification of prior information in the population and compensate for the inefficiency of information transfer within the population. Based on the dynamic multi-leader selection strategy, the whole population is separated into several bacterial clusters according to different evolution directions. Then, the multi-leaders consisting of convergence and diversity leaders are adaptively selected in each dynamic bacterial cluster. We find that these subgroups are able to continuously move towards the true Pareto front with the guidance of their convergence leader and diversity leader. In addition, we create an adaptively evolutionary elite archive for improving the convergence speed. Based on the co-evolution concept of elite archive evolution strategy, the problem of lacking sufficient searching capability due to overlapping between searching groups and elite individuals is effectively solved, whereby we can explore more objective space in the limited computational time. Besides, the new learning method is modified to match multiobjective optimization tasks and the adaptive elimination probability adjustment strategy is proposed to avoid prematurity. All these co-evolution and adaptive strategies are compatibly integrated into a comprehensive algorithm. Through the experiments on a series of well-known benchmarks with three novel MOBAs and four competitive MOEAs, MBCO/DML shows the promising performance for tackling MOPs.

TABLE 8 Spread values of MBCO/DML and four competitive MOEAs on 24 selected benchmarks.

Problem	NMPSO	BiGE	RVEA	MOCell	MBCO/DML
FON	3.4052e-1 (3.55e-2) –	1.2565e+0 (2.15e-1) –	1.5754e-1 (1.17e-2) ≈	3.9661e-1 (4.97e-2) –	1.5492e-1 (1.26e-2)
KUR	1.8784e+0 (2.12e-2) –	1.2636e+0 (1.72e-1) –	3.3203e-1 (8.92e-2) –	4.3646e-1 (4.74e-2) –	2.0532e-1 (2.11e-2)
POL	1.1960e+0 (8.94e-2) –	1.1439e+0 (1.09e-1) –	1.0120e+0 (1.62e-2) –	4.2048e-1 (5.80e-2) –	1.8655e-1 (2.01e-2)
SCH1	1.4424e+0 (9.71e-2) –	8.4813e-1 (7.69e-2) –	7.0122e-1 (1.30e-3) –	3.7669e-1 (4.85e-2) –	1.7537e-1 (1.53e-2)
SCH2	1.4303e+0 (1.51e-1) –	1.0720e+0 (5.37e-2) –	1.1288e-1 (5.52e-4) +	5.6184e-1 (4.92e-2) –	4.5059e-1 (2.39e-2)
ZDT1	1.4471e+0 (8.67e-2) –	1.0553e+0 (7.32e-2) –	2.9734e-1 (1.07e-2) –	3.2887e-1 (4.11e-2) –	1.7281e-1 (1.62e-2)
ZDT2	1.1816e+0 (6.60e-2) –	1.0865e+0 (7.81e-2) –	1.5621e-1 (2.51e-2) ≈	3.7034e-1 (4.54e-2) –	1.6079e-1 (1.80e-2)
ZDT3	1.7655e+0 (6.10e-3) –	1.1402e+0 (7.41e-2) –	2.7290e-1 (2.13e-2) –	3.8444e-1 (5.22e-2) –	2.1009e-1 (2.24e-2)
ZDT4	1.4135e+0 (1.07e-1) –	1.0445e+0 (1.24e-1) –	4.9388e-1 (1.36e-1) –	4.1596e-1 (4.96e-2) –	1.3962e-1 (1.49e-2)
ZDT6	4.8094e-1 (6.20e-2) +	1.0266e+0 (1.57e-1) ≈	2.6760e-1 (4.98e-2) +	3.7926e-1 (6.00e-2) +	7.9691e-1 (4.69e-1)
F1	1.2625e+0 (1.22e-1) –	1.0532e+0 (4.78e-2) –	7.4996e-1 (4.76e-2) –	6.8258e-1 (4.50e-2) –	1.6673e-1 (1.25e-2)
F2	1.0474e+0 (8.18e-2) –	1.0051e+0 (8.84e-3) –	9.5625e-1 (6.74e-2) –	9.5231e-1 (6.07e-2) –	1.6901e-1 (1.09e-2)
F3	1.1410e+0 (1.45e-1) –	1.0000e+0 (0.00e+0) –	9.7612e-1 (2.12e-2) –	1.0003e+0 (4.93e-4) –	8.7042e-1 (1.06e-1)
F5	1.3737e+0 (1.10e-1) –	1.0113e+0 (1.26e-2) –	8.3716e-1 (3.36e-2) –	7.8425e-1 (1.64e-2) –	1.6015e-1 (1.32e-2)
F6	1.1677e+0 (7.55e-2) –	1.0163e+0 (1.05e-2) –	9.4188e-1 (3.14e-2) –	8.7699e-1 (1.57e-2) –	1.7321e-1 (1.51e-2)
F7	1.0127e+0 (1.74e-1) –	1.0388e+0 (1.03e-1) –	9.6485e-1 (1.91e-2) ≈	1.0000e+0 (0.00e+0) –	9.5294e-1 (1.06e-1)
F9	1.5472e+0 (8.89e-2) –	1.0288e+0 (5.71e-2) –	5.7133e-1 (6.21e-2) –	4.6858e-1 (3.92e-2) –	1.8046e-1 (2.47e-2)
F10	1.0171e+0 (9.11e-3) –	1.0315e+0 (1.73e-1) –	9.8402e-1 (4.17e-2) –	9.9193e-1 (1.13e-3) –	9.5858e-1 (5.52e-3)
F4	3.3213e-1 (2.15e-2) +	1.5222e+0 (3.04e-1) –	2.2515e-1 (2.57e-2) +	8.1962e-1 (3.87e-1) –	5.8167e-1 (2.24e-1)
F8	1.1369e+0 (2.21e-1) –	1.8895e+0 (8.27e-2) –	2.2842e-1 (2.71e-2) +	8.6206e-1 (4.31e-1) –	4.9563e-1 (2.51e-1)
DTLZ4	3.4291e-1 (1.23e-1) +	8.9203e-1 (6.88e-2) –	1.9362e-1 (1.15e-1) +	5.2168e-1 (1.93e-1) +	5.5522e-1 (5.99e-2)
DTLZ5	1.0852e+0 (9.66e-2) –	1.1070e+0 (7.51e-2) –	5.6981e-1 (7.83e-2) –	3.9684e-1 (3.91e-2) –	2.4243e-1 (2.11e-2)
DTLZ6	1.0941e+0 (1.15e-1) –	1.0157e+0 (1.88e-2) –	4.7718e-1 (8.27e-2) –	5.4538e-1 (5.94e-2) –	2.3355e-1 (1.63e-2)
DTLZ7	5.6031e-1 (3.75e-2) –	9.8863e-1 (1.49e-1) –	3.6312e-1 (2.38e-2) +	4.7362e-1 (4.59e-2) +	5.1079e-1 (4.28e-2)
+/-/≈	3/21/0	0/23/1	6/15/3	3/21/0	/
Best/All	0/24	0/24	7/24	0/24	17/24

This study confirms the value and significance of the identification and transfer of prior information in population-based optimization algorithms development. In the future, more research could be devoted to the more precise identification and transmission mechanisms of population information for tackling specific tasks. In addition, the dynamic multi-leader selection mechanism will be studied further towards adaptive sub-groups gathering and robust learning for multiobjective optimization problems.

ACKNOWLEDGEMENTS

This work is supported by the National Natural Science Foundation of China (Grants Nos. 71901152, 71971143), Social Science Youth Foundation of Ministry of Education of China (21YJC630181), Special Project in Key Fields of Universities in Guangdong Province (2022ZDZX2054), Guangdong Basic and Applied Basic Research Foundation (2023A1515010919, 2023A1515011032), Guangdong Innovation Team Project of Intelligent Management and Cross Innovation (2021WCXTD002), Guangdong Provincial Philosophy and Social Sciences Planning Project (GD22XGL22), and Shenzhen Stable Support Grant (Grant Nos. 20220810100952001).

CONFLICT OF INTEREST STATEMENT

All authors declare that they have no conflict of interest.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available in PlatEMO at <https://github.com/BIMK/PlatEMO>. These data were derived from the following resources available in the public domain: PlatEMO, <https://github.com/BIMK/PlatEMO>.

ORCID

- Hong Wang  <https://orcid.org/0000-0002-4671-6122>
- Tianwei Zhou  <https://orcid.org/0000-0003-3533-7204>
- Ben Niu  <https://orcid.org/0000-0001-5822-8743>

REFERENCES

- Aboud, A., Rokbani, N., Fdhila, R., Qahtani, A. M., Almutiry, O., Dhahri, H., Hussain, A., & Alimi, A. M. (2022). DPb-MOPSO: A dynamic pareto bi-level multi-objective particle swarm optimization algorithm. *Applied Soft Computing*, 129, 109622.
- Antonio, L. M., & Coello, C. A. C. (2018). Coevolutionary multiobjective evolutionary algorithms: Survey of the state-of-the-art. *IEEE Transactions on Evolutionary Computation*, 22, 851–865.
- Beume, N., Fonseca, C. M., López-Ibáñez, M., Paquete, L., & Vahrenhold, J. (2009). On the complexity of computing the hypervolume indicator. *IEEE Transactions on Evolutionary Computation*, 13, 1075–1082.
- Bosman, P. A. N., & Thierens, D. (2003). The balance between proximity and diversity in multiobjective evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 7, 174–188.
- Cheng, R., Jin, Y., Olhofer, M., & Sendhoff, B. (2016). A reference vector guided evolutionary algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 20, 773–791.
- Deb, K., Agrawal, S., Pratap, A., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6, 182–197.
- Deb, K., Thiele, L., Laumanns, M., & Zitzler, E. (2002). Scalable multi-objective optimization test problems. In *Proceedings of the 2002 congress on evolutionary computation, CEC'02 (Cat. No. 02TH8600)* (Vol. 1, pp. 825–830).
- Dhillon, S. S., Lather, J. S., & Marwaha, S. (2016). Multi objective load frequency control using hybrid bacterial foraging and particle swarm optimized PI controller. *International Journal of Electrical Power & Energy Systems*, 79, 196–209.
- Gu, F., Liu, H., Cheung, Y.-M., & Liu, H.-L. (2023). A constrained multiobjective evolutionary algorithm based on adaptive constraint regulation. *Knowledge-Based Systems*, 260, 110112.
- Guo, C., Tang, H., & Niu, B. (2021). Evolutionary state-based novel multi-objective periodic bacterial foraging optimization algorithm for data clustering. *Expert Systems*, 39(1), e12812.
- Guo, C., Tang, H., Niu, B., & Lee, C. B. P. (2021). A survey of bacterial foraging optimization. *Neurocomputing*, 452, 728–746.
- He, Z., Hu, H., Zhang, M., Zhang, Y., & Li, A.-D. (2022). A decomposition-based multi-objective particle swarm optimization algorithm with a local search strategy for key quality characteristic identification in production processes. *Computers & Industrial Engineering*, 172, 108617.
- Kaur, M., & Kadam, S. (2018). A novel multi-objective bacteria foraging optimization algorithm (MOBFOA) for multi-objective scheduling. *Applied Soft Computing*, 66, 183–195.
- Li, J.-Y., Deng, X. Y., Zhan, Z. H., Yu, L., Tan, K. C., Lai, K. K., & Zhang, J. (2022). A multipopulation multiobjective ant colony system considering travel and prevention costs for vehicle routing in COVID-19-like epidemics. *IEEE Transactions on Intelligent Transportation Systems*, 23, 25062–25076.
- Li, M., Yang, S., & Liu, X. (2015). Bi-goal evolution for many-objective optimization problems. *Artificial Intelligence*, 228, 45–65.
- Li, M., & Zheng, J. (2009). Spread assessment for evolutionary multi-objective optimization. In *Evolutionary multi-criterion optimization* (pp. 216–230). Springer Berlin Heidelberg.
- Liao, Z., Gong, W., & Li, S. (2023). Two-stage reinforcement learning-based differential evolution for solving nonlinear equations. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 53, 4279–4290.
- Lin, Q., Liu, S., Zhu, Q., Tang, C., Song, R., Chen, J., Coello, C. A. C., Wong, K. C., & Zhang, J. (2018). Particle swarm optimization with a balanceable fitness estimation for many-objective optimization problems. *IEEE Transactions on Evolutionary Computation*, 22, 32–46.
- Liu, X.-F., Zhan, Z.-H., Gao, Y., Zhang, J., Kwong, S. T. W., & Zhang, J. (2019). Coevolutionary particle swarm optimization with bottleneck objective learning strategy for many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 23, 587–602.
- Liu, Z., Yan, J., Cheng, Q., Chu, H., Zheng, J., & Zhang, C. (2022). Adaptive selection multi-objective optimization method for hybrid flow shop green scheduling under finite variable parameter constraints: Case study. *International Journal of Production Research*, 60(12), 3844–3862.
- Ma, T.-M., Wang, X., Zhou, F.-C., & Wang, S. (2023). Research on diversity and accuracy of the recommendation system based on multi-objective optimization. *Neural Computing and Applications*, 35(7), 5155–5163.
- Martínez, S. Z., & Coello, C. A. C. (2011). A multi-objective particle swarm optimizer based on decomposition. In *Annual conference on genetic and evolutionary computation* (pp. 69–76).
- Morteza, H., Jameii, S. M., & Sohrabi, M. K. (2023). An improved learning automata based multi-objective whale optimization approach for multi-objective portfolio optimization in financial markets. *Expert Systems with Applications*, 224, 119970.
- Muller, S. D., Marchetto, J., Airaghi, S., & Kournoutsakos, P. (2002). Optimization based on bacterial chemotaxis. *IEEE Transactions on Evolutionary Computation*, 6(1), 16–29.
- Nebro, A. J., Durillo, J. J., Luna, F., Dorronsoro, B., & Alba, E. (2009). MOCell: A cellular genetic algorithm for multiobjective optimization. *International Journal of Intelligent Systems*, 24(7), 726–746.
- Niu, B., Liu, J., & Tan, L. (2019). Multi-swarm cooperative multi-objective bacterial foraging optimisation. *International Journal of Bio-Inspired Computation*, 13, 21–31.
- Niu, B., Liu, J., Wu, T., Chu, X., Wang, Z., & Liu, Y. (2018). Coevolutionary structure-redesigned-based bacterial foraging optimization. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 15, 1865–1876.
- Niu, B., Liu, Q., Wang, Z., Tan, L., & Li, L. (2020). Multi-objective bacterial colony optimization algorithm for integrated container terminal scheduling problem. *Natural Computing*, 20, 89–104.
- Niu, B., & Wang, H. (2012). Bacterial colony optimization. *Discrete Dynamics in Nature and Society*, 2012, 1–28.
- Niu, B., Wang, H., Wang, J.-W., & Tan, L. (2013). Multi-objective bacterial foraging optimization. *Neurocomputing*, 116, 336–345.

- Niu, B., Yi, W., Tan, L., Geng, S., & Wang, H. (2019). A multi-objective feature selection method based on bacterial foraging optimization. *Natural Computing*, 20, 63–76.
- Nunes, P., Moura, A., & Santos, J. (2023). Solving the multi-objective bike routing problem by meta-heuristic algorithms. *International Transactions in Operational Research*, 30(2), 717–741.
- Passino, K. M. (2012). Bacterial foraging optimization. In *Innovations and developments of swarm intelligence applications* (pp. 219–234). IGI Global.
- Pereira, J. L. J., & Gomes, G. F. (2023). Multi-objective sunflower optimization: A new hypercubic meta-heuristic for constrained engineering problems. *Expert Systems*, e13331, early view. <https://doi.org/10.1111/exsy.13331>
- Srichandan, S., Kumar, T. A., & Bibhudatta, S. L. (2018). Task scheduling for cloud computing using multi-objective hybrid bacteria foraging algorithm. *Future Computing and Informatics Journal*, 3(2), 210–230.
- Tan, L., Wang, H., Yang, C.-M., & Niu, B. (2017). A multi-objective optimization method based on discrete bacterial algorithm for environmental/economic power dispatch. *Natural Computing*, 16, 549–565.
- Tian, Y., Cheng, R., Zhang, X., Cheng, F., & Jin, Y. (2018). An indicator-based multiobjective evolutionary algorithm with reference point adaptation for better versatility. *IEEE Transactions on Evolutionary Computation*, 22, 609–622.
- Tian, Y., Cheng, R., Zhang, X., & Jin, Y. (2017). PlatEMO: A MATLAB platform for evolutionary multi-objective optimization. *IEEE Computational Intelligence Magazine*, 12, 73–87.
- Wang, D., & Cai, K. (2018). Multi-objective crashworthiness optimization of vehicle body using particle swarm algorithm coupled with bacterial foraging algorithm. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 232, 1003–1018.
- Wu, B., Hu, W., Hu, J., & Yen, G. G. (2019). Adaptive multiobjective particle swarm optimization based on evolutionary state estimation. *IEEE Transactions on Cybernetics*, 51, 3738–3751.
- Yang, K., Zheng, J., Zou, J., Yu, F., & Yang, S. (2023). A dual-population evolutionary algorithm based on adaptive constraint strength for constrained multi-objective optimization. *Swarm and Evolutionary Computation*, 77, 101247.
- Yang, Q.-T., Zhan, Z.-H., Kwong, S., & Zhang, J. (2022). Multiple populations for multiple objectives framework with bias sorting for many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 1, early access. <https://doi.org/10.1109/TEVC.2022.3212058>
- Yazdani, H., Baneshi, M., & Yaghoubi, M. (2023). Techno-economic and environmental design of hybrid energy systems using multi-objective optimization and multi-criteria decision making methods. *Energy Conversion and Management*, 282, 116873.
- Yi, J., Huang, D., Fu, S.-Y., He, H., & Li, T. (2016). Multi-objective bacterial foraging optimization algorithm based on parallel cell entropy for aluminum electrolysis production process. *IEEE Transactions on Industrial Electronics*, 63, 2488–2500.
- Zhang, Q., Zhou, A., & Jin, Y. (2008). RM-MEDA: A regularity model-based multiobjective estimation of distribution algorithm. *IEEE Transactions on Evolutionary Computation*, 12, 41–63.
- Zhang, W., Li, G., Zhang, W., Liang, J. J., & Yen, G. G. (2019). A cluster based PSO with leader updating mechanism and ring-topology for multimodal multi-objective optimization. *Swarm and Evolutionary Computation*, 50, 100569.
- Zhang, X., Zhan, Z.-H., Fang, W., Qian, P., & Zhang, J. (2021). Multipopulation ant colony system with knowledge-based local searches for multiobjective supply chain configuration. *IEEE Transactions on Evolutionary Computation*, 26, 512–526.

AUTHOR BIOGRAPHIES

Hong Wang received the bachelor's degree in Information and Computing Science from College of Information, Hainan University, Hainan, China, in 2010, the M.S. degree in Management Science and Engineering from College of Management, Shenzhen University, Shenzhen, China in 2013, and the Ph.D. degree in The Hong Kong Polytechnic University, HongKong, China in 2017. She joined the College of Management, Shenzhen University, Shenzhen, China in 2017, where she is currently a Associate Professor. Her research interests include Artificial Intelligence, Bio-inspired Computing, Multi objective Optimization, Big Data Analysis and Processing, Business Intelligence.

Yixin Wang received the B.S. degree in Industrial Engineering, from Dongguan University of Technology, Guangdong, China, in 2020. He is currently pursuing his M.S. degree in Management Science and Engineering at College of Management, Shenzhen University. His research interests focus on Intelligent optimization, Intelligent decision-making, and Learning-based algorithms.

Menglong Liu received the bachelor's degree in aircraft design and engineering and the master's degree in testing, measuring technology, and instrument from the Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2010 and 2013, respectively, and the Ph.D. degree in mechanical engineering from The Hong Kong Polytechnic University, Hong Kong, in 2017. In 2017, he joined the Institute of High Performance Computing, A*STAR, Singapore, as a Research Scientist. In 2019, he joined Harbin Institute of Technology (Shenzhen) as a tenure-track Assistant Professor in Shenzhen, China. His research interests are modeling of physical guided and bulk waves for nondestructive evaluation and structural health monitoring.

Tianwei Zhou received the B.S. degree in automation and the Ph.D. degree in control science and engineering from Tianjin University, Tianjin, China, in 2014 and 2019, respectively. She was a joint Ph.D. student with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore, from August 2017 to August 2018. She is currently an Assistant Professor with the College of Management, Shenzhen University, Shenzhen, China. Her current research interests include intelligent scheduling and event-triggered control.

Ben Niu received his Ph.D degree from Shenyang Institute of Automation of the Chinese Academy of Sciences, Shenyang, China, in 2008. In 2008, he joined the College of Management, Shenzhen University, Shenzhen, China, where he is currently a Full Professor. He was a Visiting

Scholar with the University of Edinburgh, Arizona State University, Victoria University of Wellington, Hong Kong University in 2018, 2016, 2013, and 2011. During 2013 to 2016, he worked as a Postdoctoral Research Fellow with the Hong Kong Polytechnic University. His research interests include Operations Research and Optimization, Artificial Intelligence, Big Data Analysis and Processing, Smart Healthcare, etc.

How to cite this article: Wang, H., Wang, Y., Liu, M., Zhou, T., & Niu, B. (2023). An enhanced bacterial colony optimization with dynamic multi-leader co-evolution for multiobjective optimization problems. *Expert Systems*, 40(10), e13410. <https://doi.org/10.1111/exsy.13410>