IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received December 3, 2020, accepted December 21, 2020, date of publication December 28, 2020, date of current version January 6, 2021.

Digital Object Identifier 10.1109/ACCESS.2020.3047699

Decomposition-Based Multiobjective Evolutionary Algorithm With Genetically Hybrid Differential Evolution Strategy

NAILI LUO[1,2], WU LIN[1], GENMIAO JIN[1], CHANGKUN JIANG[1], (Member, IEEE), AND JIANYONG CHEN[1]

[1]College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China
[2]College of Optoelectronic Engineering, Shenzhen University, Shenzhen 518060, China

Corresponding author: Changkun Jiang (ckjiang@szu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61876110, Grant 61902255, Grant 61836005, Grant 61672358, and Grant 61976142; in part by the Joint Funds of the National Natural Science Foundation of China under Key Program Grant U1713212; in part by the Shenzhen Technology Plan under Grant JCYJ20190808164211203 and Grant JCYJ20190808163417094; in part by the National Engineering Laboratory for Big Data System Computing Technology; and in part by the Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ), Shenzhen University.

ABSTRACT In the decomposition-based multiobjective evolutionary algorithms (MOEA/Ds), a set of subproblems are optimized by using the evolutionary search to exploit the feasible regions. In recent studies of MOEA/Ds, it was found that the design of recombination operators would significantly affect their performances. Therefore, this paper proposes a novel genetically hybrid differential evolution strategy (GHDE) for recombination in MOEA/Ds, which works effectively to strengthen the search capability. Inspired by the existing studies of recombination operators in MOEA/Ds, two composite operator pools are introduced, each of which includes two distinct differential evolution (DE) mutation strategies, one emphasizing convergence and the other focusing on diversity. Regarding each selected operator pool, two DEs are applied on parents' genes to hybridize offspring by adaptive parameters tuning. Moreover, a fitness-rate-rank-based multiarmed bandit (FRRMAB) is embedded into our algorithm to select the best operator pool by collecting their recently achieved fitness improvement rates. After embedding GHDE into an MOEA/D variant with dynamical resource allocation, a variant named MOEA/D-GHDE is presented. Various test multiobjective optimization problems (MOPs), i.e., UF, F test suites, and MOPs with difficult-to-approximate (DtA) PF boundaries, are used to assess performances. Compared to several competitive MOEA/D variants, the comprehensive experiments validate the superiority of our algorithm.

INDEX TERMS Multiobjective optimization, decomposition, recombination operator, differential evolution.

I. INTRODUCTION

Generally, a multiobjective optimization problem (MOP) is mathematically represented by

$$\text{Min } F(x) = (f_1(x), \ldots, f_m(x)), \tag{1}$$

where x is an n-dimensional decision vector in its decision space Ω, and $f_1(x), \ldots, f_m(x)$ define m objective functions [1]–[5]. The solution x is said to be a Pareto-optimal solution if and only if there is no other solution in Ω that can dominate it. Due to the mutual contradiction among different objectives, it is necessary to find more Pareto- optimal solutions (PS) for MOPs. The mappings of PS in the objective space are called the Pareto front (PF) [6]–[8].

To obtain a good approximation on PF for making decisions, multiobjective evolutionary algorithms (MOEAs) have been a popular approach to optimize various MOPs [9]–[11]. In recent decades, a variety of MOEAs have been developed to solve various theoretical optimization problems and even some real-world applications [12]–[16].

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

MOEAs are divided into several categories with different criteria in terms of the environmental selection: 1) Pareto-dominance-based framework [17]–[20], 2) indicator-based framework [21]–[23], and 3) decomposition-based framework [24]–[30].

Decomposition-based MOEAs (MOEA/Ds) decompose a MOP into multiple scalar subproblems by weight vectors, which can be optimized in a collaborative manner according to the value of the aggregate function [31]. In recent years, various strategies have been developed to further improve MOEA/Ds. In [32]–[35], a set of the weight vector adjustment strategies have been proposed. Specifically, in MOEA/D-LTD [32], the Pareto front estimated by Gaussian process (GP) regression assists to adjust weight vectors. In MOEA/D-SOM [33], weight vectors are updated by using the trained self-organizing map (SOM) network. Moreover, in AMOEA/D [34], weight vectors are adjusted by removing invalid reference points and adding some points around each crowded reference point, which can find various irregular PFs. In AWD-MOEA/D [35], the current optimal solutions are used to reset weight vectors. In addition, some resource allocation strategies have been proposed, which aim to adjust the computational resources of those subproblems with different complexities. For example, in MOEA/D-DRA [36], assigning different amounts of computational effort for one subproblem depends on the utility of the subproblem. In MOEA/D-GRA [37], a generalized resource allocation strategy is designed to associate each subproblem with a different probability according to the improvement vector. Furthermore, some efforts have been done to improve the aggregate function. For example, in eMOEA/D [38], two new aggregate functions with adjustable contours are proposed to guide the search, which can adjust the balance between diversity and convergence during evolution. In MOEA/D-LWS [39], the localized weighted sum (WS) method is introduced, which can overcome the issue that WS cannot work well for non-convex PFs.

Recombination operators including crossover and mutation are also two key components in MOEAs. For example, simulated binary crossover (SBX) is proposed in [40], which can generate offspring close to their parents. In [41], the differential evolution (DE) is another common crossover used by many MOEA/Ds. Moreover, some efforts on recombination operators have been done to further strengthen the ability of search. As introduced in [42], [43], multiple DE operators are suggested for MOEA/Ds, which can provide an enhanced search capability. Inspired by the above research works, we propose a novel genetically hybrid differential evolution strategy (GHDE) for recombination in MOEA/Ds. More specifically, four distinct differential evolution (DE) mutation strategies are classified into two composite operator pools, each of which includes two DEs. Furthermore, a fitness-rate-rank-based multiarmed bandit (FRRMAB) [42] is used to select an appropriate operator pool, in which two different DEs are applied on parents' genes to hybridize offspring by adaptive parameters tuning. In this way, our proposed algorithm works effectively to strengthen the search capability, which can realize a good balance between convergence and diversity during evolution. Our main contributions are summarized as follows:

1) A genetically hybrid differential evolution strategy is proposed, in which two DEs belonging to the same operator pool are applied on parents' genes to hybridize offspring by adaptive parameters tuning.

2) Two composite operator pools are introduced, each of which includes two distinct DE mutation strategies. Their recently achieved fitness improvement rates are collected to estimate the best operator pool.

The rest of the paper is organized as follows. In Section II, some background knowledge is introduced. In Section III, the details of MOEA/D-GHDE are described, including the genetically hybrid differential evolution strategy for recombination, the update of neighborhood solutions, and the update of parameters. In Section IV, the comparison results of our algorithm with five competitive MOEA/Ds and the performance analysis of GHDE are presented. Finally, this paper is concluded in Section V.

II. BACKGROUND KNOWLEDGE

A. DECOMPOSITION APPROACH

In MOEA/Ds, several state-of-the-art decomposition approaches can guide the subproblems to evolve [32-40]. Here, we consider using the TCH method as the aggregate function, since it is mostly used in many MOEA/Ds, as defined by

$$\min_{x\in\Omega} g^{tch}(x|\lambda^j, z^*) = \max_{1\leq i\leq m} \{|f_i(x) - z_i^*|/\lambda_i^j\}, \tag{2}$$

where $\lambda^j = (\lambda_1^j, \cdots, \lambda_m^j)^T$ is the weight vector (also the direction vector) with $\lambda_i^j \geq 0(i\ \{1,\ldots, m\})$ and $\sum_{i=1}^{m} \lambda_i = 0$. The notation z^* is an ideal point, which can be obtained by finding the minimal value of each objective.

B. OPERATOR SELECTION STRATEGY

The operator selection strategy (i.e., FRRMAB) [42] is used in this paper, including the credit assignment and the operator selection, which are shown in **Algorithm 1** and **Algorithm 2**, respectively.

Algorithm 1 Credit Assignment

Initialize each reward $Reward_i = 0$ and $n_i = 0$
for $i = 1$ **to** SlidingWindow.*length*
    $op =$ SlidingWindow.*getIndexOp*(i)
    $FIR =$ SlidingWindow.*GetFIR*(i)
    $Reward_{op} = Reward_{op} + FIR$
    $n_{op} = n_{op} + 1$
end
Rank $Reward_i$ in descending order and set $Rank_i$ to be the rank value of operator i
Compute decay value and by (4) for each *op*
Compute credit value by (5) for each *op*

Algorithm 2 The Bandit-Based Operator Selection

```
1. if There are operators that have not been selected
2.     op_t = randomly select one operator
3. else
4.
```

$$op_t = \underset{i=\{1,\ldots,K\}}{\arg\max}\left(FRR_{i,t} + C \times \sqrt{\frac{2\times\ln\sum_{j=1}^{K} n_{j,t}}{n_{i,t}}}\right)$$

```
5. end
```

In the credit assignment, the fitness improvement rates (*FIR*) of operators are reserved in a sliding window, where the length of the window is set to be W. As shown in **Algorithm 1**, each reward $Reward_i$ and n_i are set to be 0 in line 1. Then, at time point t in the sliding window, the *FIR* of operator i is computed by

$$FIR_{i,t} = \frac{pf_{i,t} - cf_{i,t}}{pf_{i,t}}, \quad (3)$$

where $pf_{i,t}$ and $cf_{i,t}$ are the fitness values of the parent and the offspring, respectively. Then, $Reward_i$ is obtained by summing all *FIR* values of operator i. In line 8, $Reward_i$ are ranked in descending order, and $Rank_i$ are the rank value of operator i. In addition, a decay value is computed by

$$Decay_i = D^{Rank_i} \times Reward_i\ , \quad (4)$$

where $D \in [0, 1]$ represents the decaying factor. Then, the credit value of operator i is computed by

$$FRR_{i,t} = \frac{Decay_i}{\sum_{j=1}^{K} Decay_j} \quad (5)$$

After obtaining the credit values, the bandit-based operator selection is used to select one suitable operator for generating new solutions as shown in **Algorithm 2**. In line 1, each operator is given an equal chance to be selected until all operators have been used at least once. After that, selecting an operator depending on *FRR*, as shown in line 4.

III. THE PROPOSED ALGORITHM

A. GENERAL FRAMEWORK

Given the above preparations, our proposed algorithm is a new MOEA/D variant based on MOEA/D-DRA [36], which is called MOEA/D-GHDE. In **Algorithm 3**, the framework of MOEA/D-GHDE is first provided in detail. To better understand its mechanism and process, its flow chart is given in Figure 1. Specifically, the algorithm starts by initializing the population, and then the recombination process is applied on parent solutions to generate offspring, including the crossover and mutation operators. Different from the traditional crossover operator, the crossover component in our algorithm includes two operator pools, each of which includes two distinct differential evolution mutation strategies. A novel genetically hybrid differential evolution strategy (i.e., **Algorithm 4**) is proposed, in which two DEs belonging to the same operator pool are applied on parents' genes to hybridize offspring by adaptive parameters tuning. In our method, FRRMAB (i.e., **Algorithm 1** and **Algorithm 2**) is used to select the best operator pool, which is applied on parents' genes to hybridize offspring. Then, the

Algorithm 3 General Framework

Input: MOP: multi-objective optimization problem
N : population size
T: the neighborhood size
δ: probability of local mating
n_r: the maximal replacement times
Maxgen: the stopping criterion
Output: S (final population)

initialize a population S with N solutions
initialize a set of weight vectors $\lambda = \{\lambda^1, \ldots, \lambda^N\}$
for $i \leftarrow 1$ to N
    $B(i) = \{i_1,\ldots, i_T\}$ where $\lambda^{i_1}, \ldots, \lambda^{i_T}$ are the T closest weight vectors to λ^i and set the utility $\pi^i = 1$
end for
initialize the reference point z^*
set $gen = 0$, $e = 0$
while the stopping criterion is not met
    select m indices of the subproblems whose objectives are respectively MOP individual objectives f_i to form set I and then select the other $\lfloor N/5 \rfloor - m$ subproblems to add into I by using 10-tournament selection based on π^i
    for each $i \in I$
        $op \leftarrow FRRMAB(FRR)$
    if $rand < \delta$
        $P \leftarrow B(i)$
    else
        $P \leftarrow \{1,\ldots, N\}$
    end if
    the scaling factor F is generated by (6) and the parameter p is adaptively adjusted by (7)
    randomly select parent solutions from S and generate y by genetically hybrid DE strategy //**Algorithm 4**
        $y \leftarrow$ Mutation ($\bar{y}$)
        evaluate y and update z^*
        update neighboring solutions //**Algorithm 5**
        $e = e+1$
    end for
    update parameters of FRRMAB by (3)-(5)
    update F' using (10)-(11)
    gen ++
    if $mod(gen, 20) == 0$
        update π^i of each subproblem using (9)
    end
end while
return S

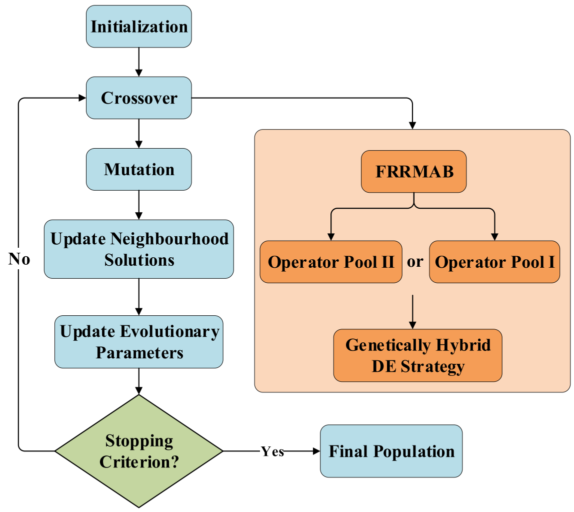

FIGURE 1. The flow chart of MOEA/D-GHDE.

promising offspring will replace at most n_r neighborhood solutions (i.e., **Algorithm 5**) and the corresponding parameters are also updated. Finally, if the stopping criterion is not reached, the evolutionary process is returned to generate offspring. Otherwise, the evolutionary process will end with the final population.

B. INITIALIZATION

First, the population S with N individuals is initialized and the weight vectors $\lambda = \{\lambda^1, \ldots, \lambda^N\}$ are evenly generated with the constraints $\sum_{i=1}^{m} \lambda_i = 1$ and $\lambda_i^j \geq 0$, where $i = \{1,\ldots, m\}$ and $j = \{1,\ldots, n\}$. After evaluating the objective functions in S, the ideal point z^* is initialized by the minimum function value of each objective. The utility π^i for subproblem i is set to be 1. Moreover, the indices of individuals in S are denoted by the set $P = \{1,2,\ldots, N\}$, while the indices of T neighbors for each i-th subproblem are marked by $B(i)(i = \{1, \ldots, N\})$, by searching the nearest weight vectors based on the Euclidean distance. The number of function evaluations (e) and the generation counter (gen) are both set as 0.

C. EVOLUTIONARY PROCESS

In line 9, as suggested by [36], a total number of $\lfloor N/5 \rfloor$ subproblems are selected for evolution and their indices are preserved in the set I, which can reasonably assign the computational resource to subproblems. First, m indices of the subproblems associated with the weight vectors (1,0,…,0),…,(0,0,…,1) are selected and the other ($\lfloor N/5 \rfloor$ -m) subproblems are randomly chosen by using the ten-tournament selection based on the utility function π. These indices are preserved in a temporary set I. For each subproblem i in I, its associated individuals will undergo the recombination procedure to generate offspring. In line 11, the operator op will be selected from {1, 2} by using FRRMAB. Then, in lines 12-16, a set of mating parents is selected as $B(i)$ with the probability δ or remains as P with the probability $(1 - \delta)$.

Algorithm 4 Genetically Hybrid DE Strategy

Input: i: the index of current subproblem
F: the scaling factor used in operator pool
p: probability parameter
parent solutions: solutions used for recombination
n: the dimension of decision variable
Output: $\bar{y}$ (offspring)
1. **switch** (op)
2. **case** 1:
3. $pool = \{$DE/rand/1, DE/current-to-rand/2$\}$
4. **for** $j \leftarrow 1$ to n
5. **if** $(rand < CR) \,||\, j == j_{rand}$
6. **if** $rand < p$
7. use DE/rand/1 to generate one gene $\bar{y}_i^j$
8. **else**
9. use DE/current-to-rand/2 to generate $\bar{y}_i^j$
10. **end if**
11. **else**
12. $\bar{y}_i^j = x_i^j$
13. **end if**
14. **end for**
15. **case** 2:
16. $pool = \{$DE/current-to-rand/1, DE/rand/2$\}$
17. **for** $j \leftarrow 1$ to n
18. **if** $(rand < CR) \,||\, j == j_{rand}$
19. **if** $rand < p$
20. use DE/current-to-rand/1 to generate $\bar{y}_i^j$
21. **else**
22. use DE/rand/2 to generate one gene $\bar{y}_i^j$
23. **end if**
24. **else**
25. $\bar{y}_i^j = x_i^j$
26. **end if**
27. **end for**
28. **end switch**
29. **return** $\bar{y}$

---

1) EVOLUTIONARY PARAMETERS

As shown in lines 17-18, before executing the crossover operator, the scaling factor F in the operator is generated by

$$F = Cauchy(F', \lambda), \tag{6}$$

where F' is the location parameter and $Cauchy(\cdot)$ represents the Cauchy distribution. The scale parameter λ is set to be 0.1 as suggested in [43].

Next, the probability parameter p can be computed as follows:

$$p = \begin{cases} rand(-0.05, 0.1), & e \leq max\_e \times 1/3 \\ 0.5, & max\_e \times 1/3 < e \leq max\_e \times 2/3 \\ rand(0.95, 1.05), & max\_e \times 2/3 < e, \end{cases} \tag{7}$$

Algorithm 5 Update Neighborhood Solutions

Input: E (neighborhood solutions)
  n_r: the number of solutions replaced
  y: offspring
Output: P (updated population)
1. $c = 1$
2. Compute η value of each solution in E by (9)
3. find x with maximal η value and replace it with y
4. $FIR_{op} = FIR_{op} + \eta(x)$
5. **while** $c < n_r$ || E != $\emptyset$
6. randomly select a solution x from E
7. **if** $\eta(x) > 0$&& $rand < \eta(x)$
8. replace x with y and delete x from E
9. $c = c$+1
10. $FIR_{op} = FIR_{op} + \eta(x)$
11. **end if**
12. **end while**
13. **return** P

where e and max_e are the current function evaluations and pre-defined maximal evaluations, respectively.

2) RECOMBINATION

The individual x_i associated with subproblem i will undergo the genetically hybrid DE strategy (**Algorithm 4**) to get $\bar{y}$. As shown in **Algorithm 4**, each composite DE operator pool consists of two different DE mutation strategies, one emphasizing convergence and the other focusing on diversity. Here, the four DE mutation strategies, i.e., DE/rand/1, DE/current-to-rand/1, DE/rand/2 and DE/current-to-rand/2 are respectively given as follows:

$$
\begin{aligned}
v^i &= x^i + F \times (x^{r_1} - x^{r_2})\\
v^i &= x^i + K \times (x^i - x^{r_1}) + F \times (x^{r_2} - x^{r_3})\\
v^i &= x^i + F \times (x^{r_1} - x^{r_2}) + F \times (x^{r_3} - x^{r_4})\\
v^i &= x^i + K \times (x^i - x^{r_1}) + F \times (x^{r_2} - x^{r_3}) + F \times (x^{r_4} - x^{r_5})
\end{aligned} \tag{8}
$$

where $x^i, x^{r_1}, x^{r_2}, x^{r_3}, x^{r_4}, x^{r_5}$ are parents and v^i is the mutated solution. The notations F and K are two parameters controlling the weighting of the difference vectors.

For each subproblem $i = 1, \ldots, n$ (n is the dimension of the decision variable), the DE mutation strategies in the selected pool are used to produce a new gene dependent on the probability parameter p. According to (7), p is randomly selected in a range of -0.05 and 0.1 when e is not larger than 1/3$*$ max_e. In this case, using the strategy DE/rand/2 or DE/current-to-rand/2 to produce a new gene has a higher probability, which can strengthen diversity. During the median period of evolution, p is set to 0.5. In this case, two distinct DE mutation strategies are applied on each gene with equal probability, which strengthens the search ability by the genetically hybrid DE strategy. At the later period, i.e., e is larger than 1/3* max_e, p is randomly selected in a range of 0.95 and 1.05. In this case, using the strategy DE/rand/1 or DE/current-to-rand/1 to produce a new gene has a higher probability, which can speed up convergence.

3) UPDATE NEIGHBORHOOD SOLUTIONS

After executing mutation on $\bar{y}$ to get y in line 19 and evaluating its objectives to update z^* in line 20, this offspring y will be used to update at most n_r individuals from S in line 21 by **Algorithm 5**. As described in **Algorithm 5**, the fitness improvement rates of all neighborhood solutions are computed by

$$
\eta(x_j) = \frac{g(x^j|\lambda^j, z^*) - g(y|\lambda^j, z^*)}{g(x^j|\lambda^j, z^*)} \tag{9}
$$

where λ^j is the weight vectors for subproblem j. First, the offspring y replaces the parent solution x with the maximal η value, and then it replaces other parent solutions until n_r parent solutions are successfully replaced. By summing up the fitness improvement rates of the successfully replaced solutions, the final reward FIR_{op} of operator op can be obtained.

Then, the number of function evaluations e is increased by 1 in line 22 of **Algorithm 3**. Each solution will undergo the above reproduction and update procedures.

D. UPDATE OF PARAMETERS

After the above reproduction and update procedures, the parameters of FRRMAB are respectively updated by (3)-(5) in line 24, which are used to select the DE operator. Then, in line 25, as suggested in [43], the location parameter F' in the Cauchy distribution can be computed by (10)-(11)

$$
\begin{aligned}
F' &= w_F \times F' + (1 - w_F)\\
&\quad \times mean_{POW}(F_{success}),
\end{aligned} \tag{10}
$$

$$
mean_{POW}(F_{success}) = \sum_{x \in F_{success}} (x^k/|F_{success}|)^{1/k}, \tag{11}
$$

where w_F is a pre-defined weight factor and $F_{success}$ is a set reserving the successful scaling factors. As suggested in [43], F' is initialized to 0.5, k is set to 1.5, and w_F is randomly sampled from [0.8, 1]. The generation counter gen is increased by 1 in line 26. At last, in lines 27-29, the utility function π of the subproblems will be recomputed at each of the 20 generations.

Finally, as shown in line 8, when the stopping criterion is reached, the population S is returned as the final solution.

IV. EXPERIMENTAL STUDIES

A. TEST PROBLEMS AND EXPERIMENTAL SETTING

In our experiments, 19 unconstrained test MOPs including UF1-10 [44] and F1-9 [45] are first used as test instances, in which their dimensions of decision variables are consistent with those in their original papers. Then, eight MOPs with difficult-to-approximate (DtA) PF boundaries [46] (MOP1_DtA_PF-MOP8_DtA_PF) are also adopted to compare the performances.

The parameters in all of the compared algorithms are set as suggested in their references, which are summarized in Table 1. In MOEA/D-GHDE, the neighborhood size

TABLE 1. Parameter settings of all compared algorithms.

Algorithms	Parameter settings
MOEA/D-DRA	T=0.1*N, δ=0.9, nr=0.01*N
MOEA/D-FRRMAB	T=20, δ=0.9, nr=2,C=5, W=0.5*N, D=1.0, CR=1.0, F=0.5 and K=0.5
MOEA/D-IR	T=20, δ=0.9, K_d =2, ϑ =8
MOEA/D-CDE	T=20, δ=0.9, nr=2, C=5, W=0.5*N
MOEA/D-MUP	T=0.1*N, δ=0.9, nr=0.1*N
MOEA/D-GHDE	T=20, δ=0.9, nr=2, CR=1.0, F=0.5 and K=0.5, $\eta = 20$, $p_m = 1/n$

$T = 20$, the probability of selecting parents in neighborhood $\delta = 0.9$, the updated size $n_r = 2$. Moreover, we set $CR = 1.0$ and $K = F$ for the DE operators, while $\eta = 20$ and $p_m = 1/n$ for the polynomial-based mutation operator.

In addition, the population size N is set to 600 for all bi-objective UF and F test problems, and to 1000 for other three-objective test problems. Moreover, we set the maximal function evaluations to 150 000 for F1-F5 and F7-F9, and to 300 000 for F6 and UF1-UF10. As suggested in [46], the population size N is set to 200, and the maximal function evaluations are set to 150 000 for MOP1_DtA_PF-MOP8_DtA_PF. To be fair, each indicator value obtained by the compared algorithms on each test problem is the averaged value over 30 independent runs.

B. PERFORMANCE INDICATORS

In this paper, we use two indicators, i.e., the inverted generational distance (IGD) [47] and the hypervolume (HV) [48], to estimate the performances of all algorithms.

1) IGD

The points uniformly sampled on the true PF is represented by P and the final solutions set obtained by an MOEA is A. Then, the IGD indicator is defined as follows:

$$IGD(P, A) = \frac{\sum_{i=1}^{|P|} d(P_i, A)}{|P|}, \qquad (12)$$

where $|P|$ is the size of P and $d(P_i\ , A)$ represents the minimal Euclidean distance in objective space from P_i to the solutions in A.

2) HV

A given reference point $z^r = (z_1^r, \ldots, z_m^r)$ in the objective space is necessary for the HV indicator to calculate the objective space size that is dominated by the solutions in S and bounded by z^r. More specifically, given z^r, the HV indicator is defined as follows:

$$HV\ (S) = VOL\left(\underset{\vec{x}\in S}{\cup}\left[f_1\ (x)\ , z_1^r\right]\times\cdots\times\left[f_m\ (x)\ , z_m^r\right]\right), \quad (13)$$

where $VOL(\cdot)$ is the Lebesgue measure. Notice that the values of the reference point in each dimension must be slightly larger than the worst value of each objective on the true PF. In our experimental settings, the reference point was set to $(1.1, 1.1)^T$ for all bi-objective problems, and to $(1.1, 1.1, 1.1)^T$ for all three-objective problems.

C. COMPARISON OF MOEA/D-GHDE AND OTHER MOEAs

The experimental results of our proposed algorithm MOEA/D-GHDE are compared to five MOEA/D variants, i.e., MOEA/D-DRA [36], MOEA/D-FRRMAB [42], MOEA/D-IR [9], MOEA/D-CDE [43], and MOEA/D-MUP [22]. Tables 2 and 3 list the mean values and the standard deviations of the two indicators, respectively, which are achieved by all the compared algorithms on UF and F instances after executing 30 independent runs. Note that the result marked by boldface indicates the best result. To show the statistical differences for the compared results obtained by MOEA/D-GHDE and the others, Wilcoxon's rank-sum test was run at a 5% significance level in our experiments. The notations "−", "+", and "∼" in Tables 2 and 3 indicate that the results obtained by the corresponding algorithm are significantly worse than, better than, and similar to those of MOEA/D-GHDE, respectively.

As observed from Table 2, MOEA/D-GHDE shows some advantages over other algorithms when considering the IGD indicator, because MOEA/D-GHDE can find the best results on 10 out of all the 19 test problems. As shown in Table 2, statistical comparisons of MOEA/D-GHDE with other competitors are summarized in the last row, in which "−/∼/+" indicates the total numbers of test problems that MOEA/D-GHDE performs better than, similarly to, and worse than the corresponding algorithm. Compared with other algorithms, MOEA/D-GHDE has some advantages because our algorithm can achieve better results on most test problems. Meanwhile, this demonstrates that our proposed GHDE plays a positive effect on solving these problems. As indicated in Table 2, MOEA/D-GHDE performs better than or similarly to MOEA/D-DRA, MOEA/D-FRRMAB, MOEA/D-IR, MOEA/D-CDE, and MOEA/D-MUP on 15, 15, 15, 14, and 18 out of 19 test problems, while it is beaten by these competitors on 4, 4, 4, 5, and 1. Therefore, according to these IGD results, we can conclude that our algorithm is better than its competitors on most problems. Recall that only one single DE operator was used in MOEA/D-DRA, MOEA/D-IR, and MOEA/D-MUP. Moreover, both MOEA/D-DRA and MOEA/D-IR adopted the adaptive resource assigning strategy, assigning different computational resources according to the relative improvement of the aggregated function values for each subproblem. However, instead of using only one ideal reference point, multiple utopian reference points were used to guide the evolutionary search directions for its subproblems in MOEA/D-MUP, aiming to explore uncovered border areas and give slight superiority in maintaining diversity without decelerating convergence. Compared with the above three algorithms, the composite DE operator pools have a positive influence on improving the performance of the proposed MOEA/D-GHDE.

TABLE 2. Experimental results on IGD obtained by all compared algorithms.

Problem	MOEA/D-DRA	MOEA/D-FRRMAB	MOEA/D-IR	MOEA/D-CDE	MOEA/D-MUP	MOEA/D-GHDE
UF1	9.87e-04(6.42e-05)+	**9.73e-04(8.27e-05)**+	1.04e-03(1.03e-04)+	1.03e-03(7.24e-05)+	1.81e-03(1.41e-04)-	1.17e-03(1.50e-04)
UF2	2.85e-03(1.47e-03)-	2.06e-03(8.13e-04)-	2.90e-03(1.44e-03)-	1.95e-03(7.80e-04)~	6.26e-03(1.82e-03)-	**1.67e-03(2.95e-04)**
UF3	7.91e-03(1.06e-02)-	3.97e-03(4.51e-03)~	1.25e-02(1.48e-02)-	5.52e-03(3.51e-03)-	8.07e-03(8.43e-03)-	**2.87e-03(2.44e-03)**
UF4	5.51e-02(3.56e-03)-	5.48e-02(4.33e-03)-	5.23e-02(3.10e-03)-	**3.53e-02(4.70e-04)**+	6.85e-02(5.45e-03)-	4.01e-02(9.62e-04)
UF5	2.86e-01(4.60e-02)-	3.07e-01(4.27e-02)-	2.59e-01(2.85e-02)-	1.90e-01(3.33e-02)-	3.39e-01(1.55e-01)-	**8.97e-02(1.66e-02)**
UF6	8.03e-02(3.67e-02)-	8.37e-02(5.74e-02)-	6.58e-02(2.36e-02)-	1.09e-01(1.61e-01)-	3.79e-01(2.15e-01)-	**5.63e-02(7.16e-03)**
UF7	1.29e-03(1.20e-03)+	1.14e-03(1.40e-04)+	**1.10e-03(1.12e-04)**+	1.17e-03(1.47e-04)+	3.36e-03(1.54e-03)-	1.46e-03(4.58e-04)
UF8	2.96e-02(4.83e-03)+	3.00e-02(5.73e-03)+	**2.35e-02(2.38e-03)**+	7.62e-02(1.62e-02)-	8.19e-02(1.73e-02)-	4.64e-02(4.88e-03)
UF9	4.86e-02(4.23e-02)-	5.12e-02(4.87e-02)-	**2.49e-02(2.10e-02)**~	5.56e-02(4.72e-02)-	1.25e-01(3.48e-02)-	2.51e-02(2.36e-02)
UF10	4.83e-01(4.23e-02)-	5.07e-01(6.98e-02)-	7.46e-01(2.29e-01)-	**3.85e-01(5.49e-02)**+	4.56e-01(6.38e-02)~	4.43e-01(8.91e-02)
F1	7.73e-04(2.10e-05)-	7.30e-04(1.09e-05)-	8.68e-04(4.11e-05)-	7.18e-04(6.71e-06)-	9.24e-04(2.47e-05)-	**6.81e-04(6.40e-06)**
F2	4.57e-02(1.69e-02)-	3.32e-02(7.51e-03)-	5.90e-02(4.63e-02)-	3.56e-02(2.74e-02)-	1.90e-02(2.69e-02)-	**4.06e-03(1.25e-03)**
F3	2.17e-02(2.01e-02)-	7.56e-03(2.97e-03)-	1.98e-02(2.89e-02)-	6.76e-03(3.25e-03)-	**5.22e-03(1.79e-03)**~	5.33e-03(4.22e-03)
F4	4.92e-03(2.55e-03)~	8.71e-03(4.74e-03)~	2.28e-02(5.47e-03)-	1.15e-02(5.55e-03)~	**2.97e-03(2.63e-04)**~	8.93e-03(8.67e-03)
F5	1.81e-02(9.63e-03)-	1.12e-02(4.07e-03)-	1.21e-02(3.60e-03)-	1.01e-02(4.36e-03)-	1.36e-02(4.80e-03)-	**6.88e-03(2.98e-03)**
F6	2.73e-02(1.48e-03)-	2.61e-02(1.39e-03)-	2.53e-02(1.27e-03)-	3.78e-02(2.13e-03)-	4.96e-02(8.08e-03)-	**2.21e-02(5.42e-04)**
F7	3.16e-01(4.78e-02)-	2.86e-01(4.76e-02)-	4.25e-01(7.72e-02)-	3.12e-01(5.95e-02)-	2.51e-01(6.48e-02)-	**1.52e-02(1.86e-02)**
F8	6.84e-02(1.43e-02)+	**5.11e-02(1.55e-02)**+	1.04e-01(5.28e-02)+	6.35e-02(1.22e-02)+	1.66e-01(1.12e-01)+	2.16e-01(2.47e-02)
F9	4.20e-02(2.72e-02)-	3.65e-02(1.03e-02)-	3.14e-02(2.04e-02)-	2.84e-02(1.84e-02)-	9.17e-03(3.68e-03)~	**8.99e-03(5.65e-03)**
Best/All	0/19	2/19	3/19	2/19	2/19	10/19
Total	14-/1~/4+	13-/2~/4+	14-/1~/4+	12-/2~/5+	14-/4~/1+	/

"-, ~, +": Competitor shows worse, similar or better performance than MOEA/D-GHDE in comparison. The best values on each test problem are marked in bold font.

Hence, we conclude that multiple DE operators can better strike a good balance between convergence and diversity during the evolutionary process of solving the test problems. Note that MOEA/D-CDE and MOEA/D-FRRMAB are also designed based on the adaptive selection of multiple DE operators. However, experimental results validate that our proposed genetically hybrid differential evolution strategy for recombination has superiority when compared with others.

Table 3 further lists the compared results on the HV indicator obtained by all compared algorithms. It turns out that similar conclusions can be obtained. As shown in Table 3, on all of the 19 test problems, the number of the best results achieved by MOEA/D-GHDE is 10, whereas other algorithms only can achieve the best performance on 0, 2, 3, 2, and 2 out of all the 19 test problems, respectively. Such an observation demonstrates that the performance of MOEA/D-GHDE is better than others on most of the test problems. As summarized in Table 3, it can be seen that on the 19 test problems, MOEA/D-GHDE still performs better than or similarly to MOEA/D-DRA, MOEA/D-FRRMAB, MOEA/D-IR, MOEA/D-CDE, and MOEA/D-MUP on 14, 14, 14, 14, and 17, whereas it is worse than these competitors on 5, 5, 5, 5, and 2, respectively. Thus, it is further confirmed by these HV results that our algorithm shows some advantages for solving most of the UF and F test problems when compared with other competitors.

To visually show the quality of the final populations obtained by different algorithms, Figures 2-3 further plot the final approximate solutions with median IGD over 30 independent runs on the test problems. Due to the page limit, only two representative cases, i.e., UF5 and F2, are given in Figures 2-3. It can be seen from Figures 2-3 that MOEA/D-GHDE have a stronger search capability, which can realize a better balance between convergence and diversity than other algorithms during evolution.

Moreover, some studies have focused on the use of statistical techniques in the analysis of the evolutionary algorithms' behaviors over optimization problems [49], [50]. To quantify the overall performance of each algorithm, Friedman's test and Bonferroni-Dunn's post hoc procedure from the software tool KEEL [50] were used to show the average performance ranks (as shown in Figure 4) and the significant differences

TABLE 3. Experimental results on HV obtained by all compared algorithms.

Problem	MOEA/D-DRA	MOEA/D-FRRMAB	MOEA/D-IR	MOEA/D-CDE	MOEA/D-MUP	MOEA/D-GHDE
UF1	8.75e-01(2.11e-04)+	**8.75e-01(2.57e-04)**+	8.74e-01(3.59e-04)+	8.75e-01(1.70e-04)+	8.74e-01(2.22e-04)-	8.74e-01(4.77e-04)
UF2	8.71e-01(2.68e-03)-	8.73e-01(1.69e-03)~	8.71e-01(2.20e-03)-	8.73e-01(1.28e-03)~	8.67e-01(2.50e-03)-	**8.73e-01(8.61e-04)**
UF3	8.62e-01(1.77e-02)-	8.70e-01(6.69e-03)~	8.54e-01(2.29e-02)-	8.67e-01(6.26e-03)-	8.64e-01(1.20e-02)-	**8.72e-01(3.41e-03)**
UF4	4.49e-01(5.64e-03)-	4.49e-01(6.83e-03)-	4.55e-01(5.10e-03)-	**4.84e-01(1.23e-03)**+	4.26e-01(8.07e-03)-	4.77e-01(1.43e-03)
UF5	1.68e-01(7.54e-02)-	1.36e-01(5.48e-02)-	1.85e-01(5.33e-02)-	3.20e-01(7.82e-02)-	2.85e-01(9.33e-02)-	**5.22e-01(3.09e-02)**
UF6	4.21e-01(6.84e-02)-	4.43e-01(4.29e-02)~	4.47e-01(4.27e-02)~	4.14e-01(7.67e-02)-	3.35e-01(1.35e-01)-	**4.58e-01(1.43e-02)**
UF7	7.07e-01(3.15e-03)+	7.07e-01(5.56e-04)+	**7.08e-01(3.10e-04)**+	7.08e-01(3.50e-04)+	7.04e-01(2.95e-03)-	7.07e-01(1.03e-03)
UF8	7.52e-01(9.69e-03)+	7.52e-01(1.08e-02)+	**7.64e-01(4.16e-03)**+	6.77e-01(1.94e-02)-	6.70e-01(2.07e-02)-	7.26e-01(8.53e-03)
UF9	1.04e+00(6.95e-02)-	1.04e+00(7.59e-02)-	**1.09e+00(3.63e-02)**+	1.01e+00(7.02e-02)-	9.17e-01(6.11e-02)-	1.08e+00(3.56e-02)
UF10	9.52e-02(3.37e-02)+	7.34e-02(3.93e-02)+	3.64e-02(3.55e-02)~	**1.66e-01(2.51e-02)**+	1.45e-01(3.98e-02)+	4.98e-02(4.85e-02)
F1	8.75e-01(6.34e-05)-	8.75e-01(6.15e-05)-	8.75e-01(7.73e-05)-	8.75e-01(2.60e-05)-	8.75e-01(5.90e-05)-	**8.76e-01(1.65e-05)**
F2	7.75e-01(2.20e-02)-	7.96e-01(1.85e-02)-	7.92e-01(4.08e-02)-	8.02e-01(4.68e-02)-	8.51e-01(3.02e-02)-	**8.69e-01(2.18e-03)**
F3	8.43e-01(2.74e-02)-	8.64e-01(5.70e-03)-	8.51e-01(2.59e-02)-	8.66e-01(5.08e-03)~	**8.69e-01(2.71e-03)**~	8.67e-01(6.00e-03)
F4	8.67e-01(4.87e-03)~	8.60e-01(7.98e-03)~	8.39e-01(7.71e-03)-	8.56e-01(9.24e-03)~	**8.72e-01(3.77e-04)**~	8.61e-01(1.49e-02)
F5	8.48e-01(1.33e-02)-	8.59e-01(5.76e-03)-	8.57e-01(5.12e-03)-	8.61e-01(6.73e-03)-	8.56e-01(9.82e-03)-	**8.65e-01(4.84e-03)**
F6	7.48e-01(3.51e-03)-	7.53e-01(3.52e-03)-	7.55e-01(2.60e-03)-	7.25e-01(4.61e-03)-	7.14e-01(1.82e-02)-	**7.69e-01(1.54e-03)**
F7	4.31e-01(5.01e-02)-	4.55e-01(4.72e-02)-	2.48e-01(8.96e-02)-	4.15e-01(6.08e-02)-	5.31e-01(6.80e-02)-	**8.28e-01(5.17e-02)**
F8	7.49e-01(2.43e-02)+	**7.78e-01(2.49e-02)**+	7.13e-01(5.24e-02)+	7.60e-01(2.01e-02)+	6.51e-01(9.73e-02)+	4.69e-01(4.22e-02)
F9	4.64e-01(3.11e-02)-	4.65e-01(2.26e-02)-	4.79e-01(2.85e-02)-	4.85e-01(4.02e-02)-	5.29e-01(6.85e-03)-	**5.32e-01(9.56e-03)**
Best/All	0/19	2/19	3/19	2/19	2/19	10/19
Total	13-/1~/5+	10-/4~/5+	12-/2~/5+	11-/3~/5+	15-/2~/2+	/

"-, ~, +": Competitor shows worse, similar or better performance than MOEA/D-GHDE in comparison. The best values on each test problem are marked in bold font.

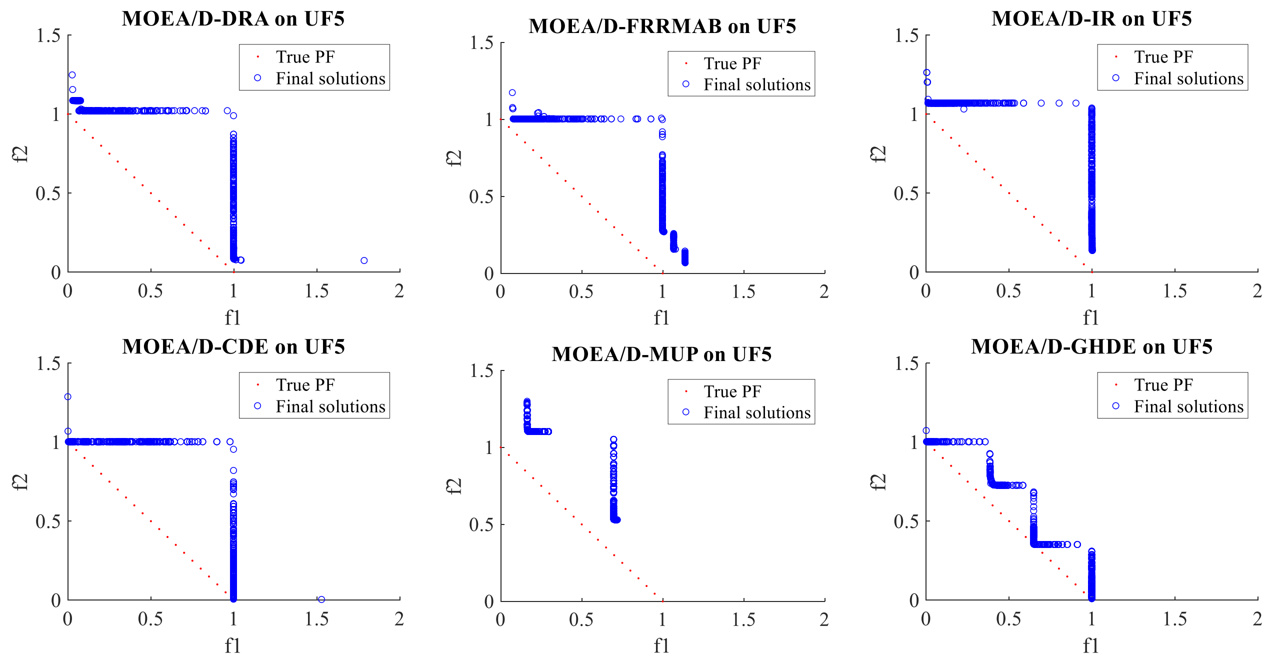

FIGURE 2. Scatter plots of population obtained by all compared algorithms on UF5.

(as shown in Table 4) between MOEA/D-GHDE and other competitors for solving UF and F test problems, respectively. From Figure 4, the average performance ranks (2.2105 in terms of IGD and 2.3684 in terms of HV) of MOEA/D-GHDE are smaller than those of the competitors, which confirms the advantages over the competitors when considering all

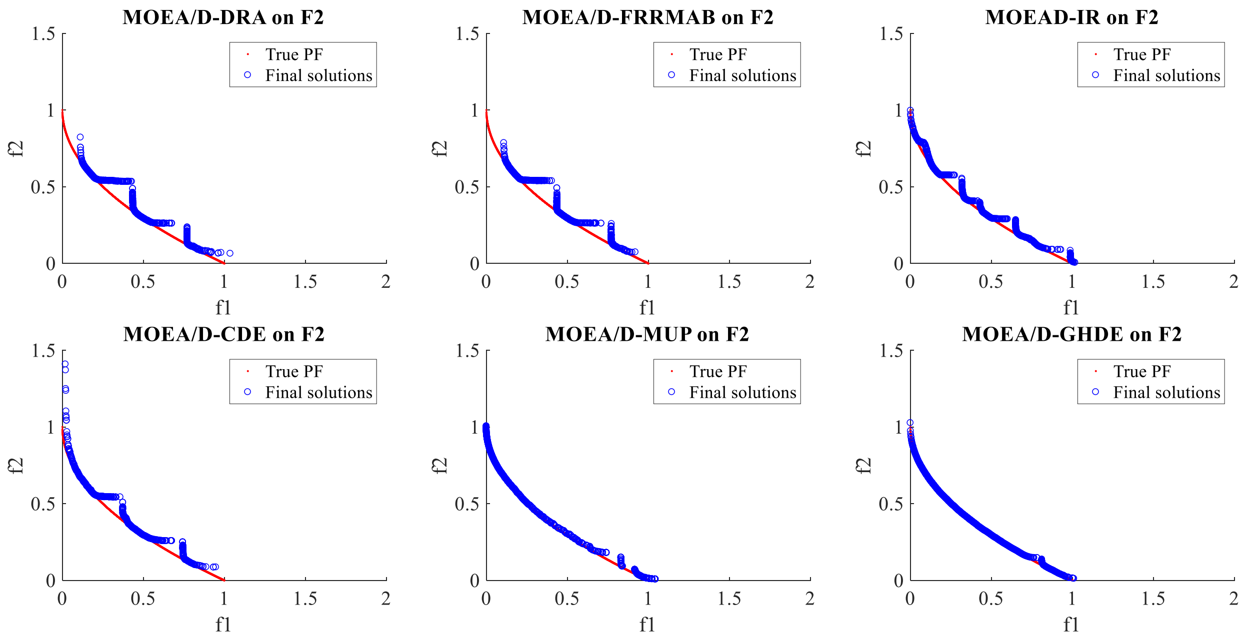

FIGURE 3. Scatter plots of population obtained by all compared algorithms on F2.

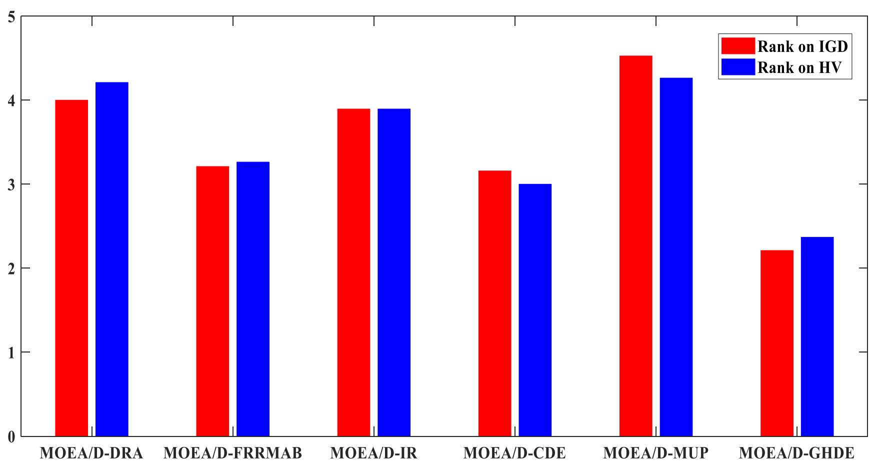

FIGURE 4. Average ranking of Friedman's test for all compared algorithms.

TABLE 4. *p*-values obtained by applying post hoc methods.

MOEA/D-GHDE vs	p-value (IGD)	p-value (HV)
MOEA/D-MUP	0.000136	0.001799
MOEA/D-DRA	0.003197	0.002406
MOEA/D-IR	0.005524	0.011916
MOEA/D-FRRMAB	0.099453	0.140458
MOEA/D-CDE	0.118571	0.298093

the test problems. Table 4 gives the post hoc comparison results, where a p-value closer to 0 means a more significant difference in the results. As described in Table 4, we can find that most p-values are very close to 0, meaning that MOEA/D-GHDE achieves more significant differences in the results.

D. EFFECTIVENESS OF GENETICALLY HYBRID DE STRATEGY

From the above experimental results, it is found that MOEA/D-GHDE shows some advantages over the competitors due to the use of the genetically hybrid differential evolution strategy. To further verify the effectiveness of the GHDE strategy, more experiments are designed to analyze the above effect in this subsection.

As introduced in Section III.C, the choice of the operator pool is decided by the parameter op, which is adjusted adaptively according to FRRMAB. Then, according to the population status, two complementary DE strategies in each operator pool are adaptively used. Here, we use two compared variants, i.e., Variant-I and Variant-II, to compare with MOEA/D-GHDE. The values of op are fixed as 1 and 2 for Variant-I and Variant-II, respectively.

Moreover, the probability parameter p is adaptively adjusted as the number of evaluations increases, which is used to realize the genetically hybrid DE strategy in each operator pool. Here, the value of p is fixed as a constant, i.e., 0.5, which is regarded as another compared variant (named Variant-III), aiming to validate the effectiveness of the adaptive strategy for adjusting p in MOEA/D-GHDE.

TABLE 5. Results of MOEA/D-GHDE and its variants on IGD.

Problem	Variant-I	Variant-II	Variant-III	MOEA/D-GHDE
UF1	1.37e-03(1.76e-04)-	1.23e-03(2.39e-04)~	3.15e-03(4.57e-04)-	1.17e-03(1.50e-04)
UF2	1.74e-03(3.10e-04)~	1.70e-03(3.36e-04)~	2.35e-03(9.56e-04)-	1.67e-03(2.95e-04)
UF3	2.15e-03(1.34e-03)~	4.40e-03(4.71e-03)-	1.23e-02(1.09e-02)-	2.87e-03(2.44e-03)
UF4	4.19e-02(1.54e-03)-	3.98e-02(1.04e-03)~	4.08e-02(1.45e-03)-	4.01e-02(9.62e-04)
UF5	9.58e-02(1.74e-02)~	8.24e-02(7.30e-03)+	8.34e-02(9.45e-03)+	8.97e-02(1.66e-02)
UF6	5.71e-02(6.53e-03)~	6.42e-02(1.84e-02)-	6.27e-02(2.43e-03)-	5.63e-02(7.16e-03)
UF7	1.57e-03(2.10e-04)-	1.76e-03(6.27e-04)-	3.13e-03(6.00e-04)-	1.46e-03(4.58e-04)
UF8	4.30e-02(5.58e-03)+	6.33e-02(1.86e-02)-	6.62e-02(9.69e-03)-	4.64e-02(4.88e-03)
UF9	3.18e-02(2.88e-02)-	2.53e-02(2.59e-02)~	3.36e-02(1.99e-03)-	2.51e-02(2.36e-02)
UF10	5.64e-01(1.25e-01)-	4.59e-01(1.38e-01)~	1.18e+00(3.41e-01)-	4.43e-01(8.91e-02)
F1	6.79e-04(5.77e-06)~	6.89e-04(1.40e-05)-	7.09e-04(1.40e-05)-	6.81e-04(6.40e-06)
F2	7.50e-03(4.47e-03)-	4.97e-03(1.20e-03)-	1.68e-02(7.13e-03)-	4.06e-03(1.25e-03)
F3	3.97e-03(1.92e-03)~	4.10e-03(2.70e-03)~	6.40e-03(5.16e-03)~	5.33e-03(4.22e-03)
F4	1.16e-02(8.09e-03)-	1.34e-03(3.11e-04)+	1.26e-02(9.71e-03)-	8.93e-03(8.67e-03)
F5	7.71e-03(3.40e-03)~	8.27e-03(2.91e-03)-	7.10e-03(2.55e-03)~	6.88e-03(2.98e-03)
F6	2.21e-02(4.76e-04)~	2.28e-02(1.03e-03)-	2.84e-02(9.42e-04)-	2.21e-02(5.42e-04)
F7	9.33e-03(1.74e-02)~	3.71e-02(2.83e-02)-	1.52e-01(4.89e-02)-	1.52e-02(1.86e-02)
F8	1.95e-01(3.72e-02)+	2.60e-01(4.96e-02)-	3.72e-01(6.19e-02)-	2.16e-01(2.47e-02)
F9	8.79e-03(2.77e-03)~	9.76e-03(3.30e-03)~	2.05e-02(8.70e-03)-	8.99e-03(5.65e-03)
Total	7-/10~/2+	10-/7~/2+	16-/2~/1+	/

"-, ~, +": Competitor shows worse, similar or better performance than MOEA/D-GHDE in comparison.

In this experiment, the performances of the three variants are tested on 19 test problems including UF1-10 and F1-9. The averaged results regarding the IGD and the HV indicators over 30 independent runs achieved by MOEA/D-GHDE, Variant-I, Variant-II, and Variant-III are given in Tables 5 and 6, respectively. Specifically, as shown in Table 5, when considering the IGD indicator, MOEA/D-GHDE performs better than or similarly to Variant-I, Variant-II, and Variant-III on 17, 17, and 18 out of 19 test problems, respectively. Moreover, as shown in Table 6, when considering the HV indicator, MOEA/D-GHDE performs better than or similarly to Variant-I, Variant-II, and Variant-III, on 19, 16, and 19 out of 19 test problems, respectively. Therefore, the superior performance of MOEA/D-GHDE over the three variants further validates the effectiveness of the GHDE strategy.

TABLE 6. Results of MOEA/D-GHDE and its variants on HV.

Problem	Variant-I	Variant-II	Variant-III	MOEA/D-GHDE
UF1	8.74e-01(3.85e-04)-	8.74e-01(7.12e-04)-	8.71e-01(1.02e-03)-	8.74e-01(4.77e-04)
UF2	8.73e-01(5.36e-04)~	8.73e-01(6.96e-04)~	8.72e-01(1.72e-03)-	8.73e-01(8.61e-04)
UF3	8.73e-01(1.89e-03)~	8.68e-01(1.39e-02)-	8.55e-01(1.97e-02)-	8.72e-01(3.41e-03)
UF4	4.73e-01(2.46e-03)-	4.77e-01(1.16e-03)~	4.76e-01(1.68e-03)-	4.77e-01(1.43e-03)
UF5	5.11e-01(3.22e-02)~	5.36e-01(1.36e-02)+	5.33e-01(1.80e-02)~	5.22e-01(3.09e-02)
UF6	4.56e-01(1.44e-02)~	4.45e-01(1.69e-02)-	4.38e-01(4.48e-03)-	4.58e-01(1.43e-02)
UF7	7.06e-01(6.56e-04)-	7.06e-01(1.31e-03)-	7.04e-01(1.30e-03)-	7.07e-01(1.03e-03)
UF8	7.27e-01(8.18e-03)~	6.95e-01(2.02e-02)-	6.69e-01(1.74e-02)-	7.26e-01(8.53e-03)
UF9	1.07e+00(4.90e-02)-	1.08e+00(3.57e-02)~	1.05e+00(7.40e-03)-	1.08e+00(3.56e-02)
UF10	3.15e-02(3.74e-02)~	7.21e-02(5.22e-02)~	1.15e-03(3.92e-03)-	4.98e-02(4.85e-02)
F1	8.76e-01(3.25e-05)-	8.76e-01(3.08e-05)~	8.76e-01(2.72e-05)-	8.76e-01(1.65e-05)
F2	8.62e-01(1.04e-02)-	8.67e-01(1.92e-03)-	8.45e-01(1.80e-02)-	8.69e-01(2.18e-03)
F3	8.70e-01(3.17e-03)~	8.69e-01(4.38e-03)~	8.66e-01(7.16e-03)~	8.67e-01(6.00e-03)
F4	8.56e-01(1.39e-02)-	8.74e-01(8.19e-04)+	8.55e-01(1.60e-02)-	8.61e-01(1.49e-02)
F5	8.64e-01(5.64e-03)~	8.63e-01(3.64e-03)-	8.64e-01(4.84e-03)~	8.65e-01(4.84e-03)
F6	7.67e-01(1.13e-03)-	7.67e-01(2.34e-03)-	7.51e-01(2.40e-03)-	7.69e-01(1.54e-03)
F7	8.49e-01(4.82e-02)~	7.71e-01(7.27e-02)-	5.62e-01(9.49e-02)-	8.28e-01(5.17e-02)
F8	4.70e-01(8.27e-02)~	3.54e-01(6.91e-02)-	1.88e-01(6.74e-02)-	4.69e-01(4.22e-02)
F9	5.31e-01(5.76e-03)~	5.33e-01(1.39e-03)+	5.12e-01(1.88e-02)-	5.32e-01(9.56e-03)
Total	8-/11~/0+	10-/6~/3+	16-/3~/0+	/

"-, ~, +": Competitor shows worse, similar or better performance than MOEA/D-GHDE in comparison.

E. PARAMETER SENSITIVITY ANALYSIS

To capture the impact of the evolutionary parameters, the parameter sensitivity analyses of the scale parameter λ and the probability parameter p are studied, respectively.

First of all, keeping the same parameter setting as introduced in the previous experiments, MOEA/D-GHDE with different λ values from {0.05, 0.1, 0.2, 0.3} were experimentally compared. As shown in Tables 7 and 8, MOEA/D-GHDE with $\lambda = 0.1$ achieves better performances on most test problems when comparing to MOEA/D-GHDE with other λ values. More specifically, regarding IGD, MOEA/D-GHDE with $\lambda = 0.1$ achieves better or similar results than those using $\lambda = 0.05$, 0.2, and 0.3 on 17, 18, 18 out of 19 cases, while only worse than its competitors on 2, 1, and 1. Moreover, regarding HV, MOEA/D-GHDE with $\lambda = 0.1$ achieves better or similar results on 16, 17, 18 out of 19 cases, while only worse than its competitors on 3, 2, and 1. Therefore, the above comparison results validate the effectiveness of MOEA/D-GHDE with $\lambda = 0.1$.

Then, to capture the impact of the probability parameter p on the performance, also keeping the same parameter setting as before, MOEA/D-GHDE with different p values

TABLE 7. Experimental results on IGD obtained by MOEA/D-GHDE with different λ and *p* values.

Problem	λ=0.05	λ=0.2	λ=0.3	*p*=0.3	*p*=0.8	MOEA/D-GHDE
UF1	1.25e-03(4.89e-04)~	1.12e-03(1.08e-04)~	1.14e-03(1.16e-04)~	1.26e-03(3.94e-04)~	1.15e-03(1.30e-04)~	1.17e-03(1.50e-04)
UF2	1.79e-03(4.42e-04)~	1.81e-03(4.19e-04)~	1.63e-03(2.55e-04)~	1.67e-03(2.52e-04)~	1.81e-03(7.37e-04)~	1.67e-03(2.95e-04)
UF3	5.21e-03(4.22e-03)-	3.75e-03(2.94e-03)~	4.56e-03(4.43e-03)-	4.03e-03(3.29e-03)~	4.91e-03(4.54e-03)-	2.87e-03(2.44e-03)
UF4	4.02e-02(1.13e-03)~	4.03e-02(8.99e-04)~	4.03e-02(1.30e-03)~	4.05e-02(1.23e-03)~	4.03e-02(1.59e-03)~	4.01e-02(9.62e-04)
UF5	8.35e-02(6.77e-03)~	8.57e-02(1.49e-02)~	8.74e-02(2.61e-02)~	8.39e-02(8.88e-03)~	8.37e-02(1.10e-02)~	8.97e-02(1.66e-02)
UF6	6.04e-02(5.40e-03)-	6.09e-02(1.23e-02)~	6.26e-02(1.01e-02)-	6.45e-02(2.49e-02)-	6.05e-02(8.45e-03)-	5.63e-02(7.16e-03)
UF7	2.01e-03(6.89e-04)-	1.86e-03(4.60e-04)-	1.86e-03(6.11e-04)-	1.86e-03(6.29e-04)-	1.72e-03(4.64e-04)-	1.46e-03(4.58e-04)
UF8	7.09e-02(2.25e-02)-	6.72e-02(2.06e-02)-	6.43e-02(1.71e-02)-	6.91e-02(2.24e-02)-	5.91e-02(1.54e-02)-	4.64e-02(4.88e-03)
UF9	2.06e-02(4.33e-04)~	2.06e-02(5.75e-04)~	2.06e-02(6.30e-04)~	2.06e-02(4.36e-04)~	2.05e-02(4.40e-04)~	2.51e-02(2.36e-02)
UF10	3.95e-01(9.91e-02)+	4.75e-01(1.48e-01)~	4.25e-01(8.52e-02)~	4.08e-01(1.20e-01)+	4.50e-01(1.39e-01)~	4.43e-01(8.91e-02)
F1	6.88e-04(1.46e-05)~	6.89e-04(1.36e-05)-	6.89e-04(1.43e-05)~	6.91e-04(1.41e-05)-	6.90e-04(1.42e-05)-	6.81e-04(6.40e-06)
F2	5.28e-03(1.69e-03)-	4.88e-03(1.52e-03)-	4.94e-03(1.36e-03)-	5.49e-03(2.20e-03)-	5.97e-03(2.31e-03)-	4.06e-03(1.25e-03)
F3	4.69e-03(2.73e-03)~	4.02e-03(2.46e-03)~	4.58e-03(2.56e-03)~	4.44e-03(3.00e-03)~	3.74e-03(2.43e-03)~	5.33e-03(4.22e-03)
F4	1.28e-03(3.00e-04)+	1.35e-03(5.84e-04)+	1.24e-03(1.45e-04)+	1.42e-03(5.17e-04)+	1.93e-03(3.73e-03)+	8.93e-03(8.67e-03)
F5	8.45e-03(3.06e-03)-	8.35e-03(2.36e-03)-	9.46e-03(3.30e-03)-	8.12e-03(1.89e-03)-	7.48e-03(1.81e-03)~	6.88e-03(2.98e-03)
F6	2.27e-02(7.12e-04)-	2.28e-02(8.63e-04)-	2.26e-02(4.85e-04)-	2.27e-02(7.15e-04)-	2.27e-02(8.46e-04)-	2.21e-02(5.42e-04)
F7	3.77e-02(2.58e-02)-	4.24e-02(2.62e-02)-	2.76e-02(2.61e-02)-	3.71e-02(3.12e-02)-	2.86e-02(2.97e-02)~	1.52e-02(1.86e-02)
F8	2.60e-01(4.10e-02)-	2.45e-01(4.86e-02)-	2.60e-01(4.03e-02)-	2.34e-01(4.84e-02)-	2.42e-01(5.50e-02)-	2.16e-01(2.47e-02)
F9	8.39e-03(2.63e-03)~	8.83e-03(2.80e-03)~	1.04e-02(3.69e-03)-	9.78e-03(3.38e-03)~	1.14e-02(4.66e-03)-	8.99e-03(5.65e-03)
Total	9-/8~/2+	8-/10~/1+	10-/8~/1+	9-/8~/2+	9-/9~/1+	/

from {0.3, 0.5, 0.8} were compared. The results show that MOEA/D-GHDE with $p = 0.5$ achieves better IGD results than those using other p values ($p = 0.3$ and 0.8) on 9 and 9 out of 19 cases, while it is better than its competitors on 11 and 9 regarding HV. Moreover, MOEA/D-GHDE with $p = 0.5$ is only worse than its competitors on 2 and 1 regarding IGD, while on 3 and 2 regarding HV. Based on these observations, we can conclude that MOEA/D-GHDE with $p = 0.5$ is more reasonable than other values.

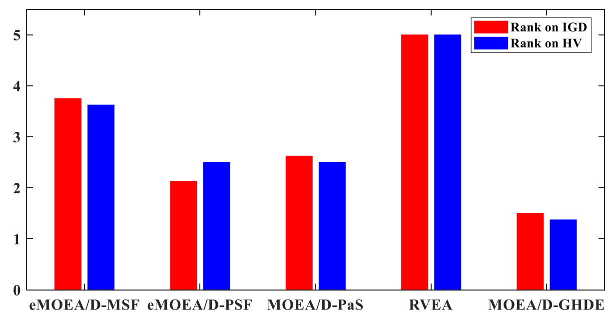

FIGURE 5. Average ranking of Friedman's test.

F. COMPETITION WITH OTHER RECENT MOEAs

In this subsection, another eight test instances featured by difficult-to-approximate (DtA) PF boundaries [46] are adopted to further investigate the performance of MOEA/D-GHDE and four popular or recently proposed MOEAs (i.e., eMOEA/D-MSF [38], eMOEA/D-PSF [38], MOEA/D-PaS [51] and RVEA [52]. Notice that the settings of the four compared MOEAs are suggested in their references. Tables 9 and 10 give the experimental results of MOEA/D-GHDE with the four MOEAs regarding IGD and HV, respectively. In Tables 9 and 10, "+/–/∼" indicate that the competitors performed better than, worse than, and similar to MOEA/D-GHDE, respectively.

The results show that the IGD and HV values achieved by eMOEA/D-MSF and RVEA are always worse than those achieved by MOEA/D-GHDE. Compared with eMOEA/D-PSF regarding IGD, MOEA/D-GHDE performs better than eMOEA/D-PSF on 6 cases, and achieves 1 similar case and 1 worse case, respectively. Similarly, regarding HV, MOEA/D-GHDE achieves significantly better and similar results than eMOEA/D-PSF on 7 cases and 1 case, respectively. The above comparison results confirm that MOEA/D-GHDE performs better than eMOEA/D-MSF, eMOEA/D-PSF, and RVEA. Moreover, MOEA/D-GHDE and MOEA/D-PaS achieve similar performances on these problems, demonstrating the effectiveness of the Pareto adaptive scalarizing (PaS) approximation method proposed in

TABLE 8. Experimental results on HV obtained by MOEA/D-GHDE with different λ and p values.

Problem	λ=0.05	λ=0.2	λ=0.3	p=0.3	p=0.8	MOEA/D-GHDE
UF1	8.74e-01(7.33e-04)~	8.74e-01(4.29e-04)~	8.74e-01(4.49e-04)~	8.74e-01(6.92e-04)-	8.74e-01(4.60e-04)~	8.74e-01(4.77e-04)
UF2	8.73e-01(8.80e-04)~	8.73e-01(6.93e-04)~	8.73e-01(7.22e-04)~	8.73e-01(6.09e-04)~	8.73e-01(1.54e-03)~	**8.73e-01(8.61e-04)**
UF3	8.68e-01(6.71e-03)-	8.70e-01(7.35e-03)~	8.69e-01(8.55e-03)-	8.69e-01(7.20e-03)-	8.68e-01(8.13e-03)-	**8.72e-01(3.41e-03)**
UF4	4.77e-01(1.41e-03)~	4.76e-01(1.59e-03)~	4.76e-01(1.78e-03)~	4.76e-01(1.43e-03)~	4.77e-01(1.82e-03)~	4.77e-01(1.43e-03)
UF5	5.34e-01(1.30e-02)~	5.29e-01(2.90e-02)~	5.26e-01(5.42e-02)~	5.33e-01(1.62e-02)~	5.34e-01(2.07e-02)~	**5.22e-01(3.09e-02)**
UF6	4.49e-01(1.33e-02)-	4.54e-01(1.72e-02)~	4.47e-01(1.34e-02)-	4.49e-01(1.35e-02)-	4.50e-01(1.35e-02)-	**4.58e-01(1.43e-02)**
UF7	7.06e-01(1.46e-03)-	7.06e-01(9.63e-04)-	7.06e-01(1.21e-03)-	7.06e-01(1.21e-03)-	7.06e-01(9.91e-04)-	7.07e-01(1.03e-03)
UF8	6.84e-01(1.93e-02)-	6.90e-01(1.92e-02)-	6.89e-01(1.79e-02)-	6.88e-01(2.04e-02)-	6.94e-01(1.88e-02)-	7.26e-01(8.53e-03)
UF9	1.09e+00(2.42e-03)~	1.09e+00(2.28e-03)~	1.09e+00(2.12e-03)~	1.09e+00(2.44e-03)+	1.09e+00(2.19e-03)+	1.08e+00(3.56e-02)
UF10	9.09e-02(4.09e-02)+	7.74e-02(4.69e-02)+	7.53e-02(4.91e-02)~	9.24e-02(4.29e-02)+	7.97e-02(5.37e-02)~	4.98e-02(4.85e-02)
F1	8.76e-01(3.79e-05)~	8.76e-01(2.85e-05)~	8.76e-01(2.81e-05)~	8.76e-01(3.18e-05)~	8.76e-01(2.95e-05)-	**8.76e-01(1.65e-05)**
F2	8.67e-01(2.69e-03)-	8.67e-01(2.45e-03)-	8.67e-01(2.18e-03)-	8.67e-01(3.44e-03)-	8.66e-01(3.87e-03)-	**8.69e-01(2.18e-03)**
F3	8.67e-01(4.04e-03)~	8.68e-01(3.98e-03)~	8.68e-01(4.28e-03)~	8.68e-01(5.09e-03)~	8.69e-01(3.89e-03)~	8.67e-01(6.00e-03)
F4	8.74e-01(8.61e-04)+	8.74e-01(1.38e-03)+	8.74e-01(4.49e-04)+	8.74e-01(1.32e-03)+	8.73e-01(6.32e-03)+	8.61e-01(1.49e-02)
F5	8.62e-01(3.34e-03)-	8.62e-01(3.24e-03)-	8.61e-01(3.22e-03)-	8.62e-01(2.53e-03)-	8.63e-01(2.46e-03)~	**8.65e-01(4.84e-03)**
F6	7.67e-01(1.81e-03)-	7.67e-01(2.65e-03)-	7.67e-01(1.52e-03)-	7.67e-01(1.91e-03)-	7.66e-01(1.50e-03)-	**7.69e-01(1.54e-03)**
F7	7.67e-01(6.87e-02)-	7.56e-01(6.47e-02)-	7.96e-01(6.77e-02)-	7.73e-01(7.87e-02)-	7.96e-01(7.77e-02)~	**8.28e-01(5.17e-02)**
F8	3.62e-01(5.89e-02)-	3.82e-01(9.42e-02)-	3.60e-01(5.82e-02)-	3.96e-01(7.47e-02)-	3.73e-01(8.55e-02)-	4.69e-01(4.22e-02)
F9	5.32e-01(2.17e-03)+	5.32e-01(5.08e-03)-	5.32e-01(5.29e-03)-	5.32e-01(3.93e-03)-	5.31e-01(6.80e-03)-	**5.32e-01(9.56e-03)**
Total	9-/7~/3+	8-/9~/2+	10-/8~/1+	11-/5~/3+	9-/8~/2+	/

"-, ~, +": Competitor shows worse, similar or better performance than MOEA/D-GHDE in comparison.

TABLE 9. Experimental results on IGD values obtained by the recent MOEAs and MOEA/D-GHDE.

Problem	eMOEA/D-MSF	eMOEA/D-PSF	MOEA/D-PaS	RVEA	MOEA/D-GHDE
MOP1-DtA_PF	4.20e-02(4.22e-03)-	3.11e-02(3.33e-03)-	2.50e-02(2.99e-03)~	6.62e-02(4.85e-03)-	2.42e-02(4.91e-03)
MOP2-DtA_PF	4.83e-02(4.46e-03)-	3.56e-02(2.94e-03)-	3.11e-02(2.76e-03)~	7.09e-02(4.18e-03)-	3.16e-02(4.11e-03)
MOP3-DtA_PF	7.89e-02(6.21e-03)-	5.42e-02(4.32e-03)-	4.74e-02(3.11e-03)~	1.10e-01(1.37e-02)-	4.46e-02(1.21e-02)
MOP4-DtA_PF	8.65e-02(6.00e-03)-	6.29e-02(4.50e-03)-	6.33e-02(5.29e-02)-	1.22e-01(1.20e-02)-	5.69e-02(3.48e-03)
MOP5-DtA_PF	1.32e-01(7.51e-03)-	9.23e-02(9.56e-03)-	1.38e-01(1.08e-01)-	1.87e-01(2.55e-02)-	6.36e-02(3.24e-02)
MOP6-DtA_PF	1.37e-01(8.30e-03)-	9.96e-02(1.06e-02)-	1.85e-01(1.29e-01)~	2.05e-01(1.53e-02)-	9.28e-02(1.06e-02)
MOP7-DtA_PF	2.36e-01(1.17e-02)-	2.03e-01(1.41e-02)~	2.24e-01(7.71e-02)~	2.52e-01(1.25e-02)-	2.08e-01(9.50e-03)
MOP8-DtA_PF	1.21e-01(1.26e-02)-	7.02e-02(1.94e-02)+	8.00e-02(4.00e-02)+	1.53e-01(9.01e-03)-	9.03e-02(1.21e-02)
Total	8-/0~/0+	6-/1~/1+	2-/5~/1+	8-/0~/0+	/

"-, ~, +": Competitor shows worse, similar or better performance than MOEA/D-GHDE in comparison.

MOEA/D-PaS. However, although MOEA/D-GHDE adopts the traditional scalarizing method, it still achieves similar performances on most problems, showing the effectiveness of using the proposed genetically hybrid differential evolution strategy.

In addition, we use the non-parametric tests for analyzing the performances of the compared algorithms on these test problems. As shown in Figure. 5, the average performance ranks of all compared algorithms are computed by using Friedman's test method. We can find that the average performance ranks (1.5 on IGD and 1.4375 on HV) of MOEA/D-GHDE are smaller than those of other MOEAs. Moreover, the p-values obtained from Bonferroni-Dunn's and Holm's post hoc procedure are provided in Table 11. Most p-values are close to 0, showing that the results of the proposed MOEA/D-GHDE are significantly better than the competitors when solving these test problems.

TABLE 10. Experimental results on HV values obtained by the recent MOEAs and MOEA/D-GHDE.

Problem	eMOEA/D-MSF	eMOEA/D-PSF	MOEA/D-PaS	RVEA	MOEA/D-GHDE
MOP1_DtA_PF	5.25e-01(3.47e-03)-	5.34e-01(3.63e-03)-	5.42e-01(4.68e-03)+	4.94e-01(3.45e-03)-	5.39e-01(2.26e-16)
MOP2_DtA_PF	5.19e-01(2.89e-03)-	5.29e-01(2.93e-03)~	5.34e-01(3.28e-03)+	4.90e-01(4.02e-03)-	5.29e-01(3.39e-16)
MOP3_DtA_PF	4.91e-01(2.92e-03)-	5.07e-01(3.67e-03)-	5.14e-01(2.75e-03)+	4.54e-01(1.08e-02)-	5.10e-01(4.52e-16)
MOP4_DtA_PF	4.84e-01(3.47e-03)-	4.99e-01(3.31e-03)-	5.00e-01(4.37e-02)-	4.45e-01(8.03e-03)-	5.03e-01(0.00e+00)
MOP5_DtA_PF	4.48e-01(3.92e-03)-	4.69e-01(7.34e-03)-	4.31e-01(9.05e-02)~	3.97e-01(1.82e-02)-	4.77e-01(1.69e-16)
MOP6_DtA_PF	4.43e-01(4.45e-03)-	4.63e-01(8.11e-03)-	3.95e-01(1.03e-01)-	3.85e-01(1.02e-02)-	4.72e-01(1.69e-16)
MOP7_DtA_PF	3.55e-01(4.36e-03)-	3.67e-01(5.48e-03)-	3.52e-01(6.79e-02)~	3.47e-01(6.76e-03)-	3.75e-01(5.65e-17)
MOP8_DtA_PF	7.96e-01(2.76e-03)-	8.05e-01(3.16e-03)-	8.04e-01(4.97e-02)~	7.68e-01(4.21e-03)-	8.12e-01(0.00e+00)
Total	8-/0~/0+	7-/1~/0+	2-/3~/3+	8-/0~/0+	/

"-, ~, +": Competitor shows worse, similar or better performance than MOEA/D-GHDE in comparison.

TABLE 11. *p*-values obtained by using post hoc methods.

MOEA/D-GHDE vs	*p*-value (IGD)	*p*-value (HV)
RVEA	0.00001	0.000007
eMOEA/D-MSF	0.004427	0.005658
eMOEA/D-PaS	0.154729	0.178959
eMOEA/D-PSF	0.429195	0.205903

V. CONCLUSION AND FUTURE WORK

In this paper, a decomposition-based multiobjective evolutionary algorithm with a novel genetically hybrid differential evolution strategy (GHDE) for recombination is proposed. Two composite operator pools are designed in our algorithm, each of which includes two DE mutation strategies. Moreover, the selection of two composite operator pools is decided by FRRMAB. In each composite operator pool, two different DEs are applied on parents' genes to hybridize offspring by adaptive parameters tuning. Finally, 19 complex test MOPs are adopted for performance comparison, and the results show that our proposed algorithm exhibits the superiorities over five competitive MOEA/D variants (MOEA/D-DRA, MOEA/D-FRRMAB, MOEA/D-IR, MOEA/D-CDE, and MOEA/D-MUP). Meanwhile, the effectiveness of the proposed GHDE strategy is confirmed and the parameters sensitivities are further analyzed on these test problems. Furthermore, eight MOPs with difficult-to-approximate PF boundaries are adopted to further verify the superior performance of MOEA/D-GHDE when compared with some popular or recently proposed MOEAs (i.e., eMOEA/D-MSF, eMOEA/D-PSF, MOEA/D-PaS, and RVEA.).

In future work, more efficient adaptive operator selection strategies will be studied to further strengthen the search ability in MOEA/Ds. Moreover, it is interesting to extend the proposed algorithm MOEA/D-GHDE to solve more realistic multiobjective optimization problems, even some real-world applications [53]–[55].

REFERENCES

[1] K. Miettinen, *Nonlinear Multiobjective Optimization*. Boston, MA, USA: Kluwer Academic, 1999.

[2] Y. Shen and G. Ge, "Multi-objective particle swarm optimization based on fuzzy optimality," *IEEE Access*, vol. 7, pp. 101513–101526, 2019, doi: 10.1109/ACCESS.2019.2926584.

[3] Z. He, G. G. Yen, and Z. Yi, "Robust multiobjective optimization via evolutionary algorithms," *IEEE Trans. Evol. Comput.*, vol. 23, no. 2, pp. 316–330, Apr. 2019.

[4] Q. Feng, Q. Li, P. Chen, H. Wang, Z. Xue, L. Yin, and C. Ge, "Multiobjective particle swarm optimization algorithm based on adaptive angle division," *IEEE Access*, vol. 7, pp. 87916–87930, 2019, doi: 10.1109/ACCESS.2019.2925540.

[5] C. Zhu, L. Xu, and E. D. Goodman, "Generalization of Pareto-optimality for many-objective evolutionary optimization," *IEEE Trans. Evol. Comput.*, vol. 20, no. 2, pp. 299–315, Apr. 2016.

[6] Z. Zhu, X. Tian, C. Xia, L. Chen, and Y. Cai, "A shift vector guided multiobjective evolutionary algorithm based on decomposition for dynamic optimization," *IEEE Access*, vol. 8, pp. 38391–38403, 2020, doi: 10.1109/ACCESS.2020.2974324.

[7] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: Empirical results," *Evol. Comput.*, vol. 8, no. 2, pp. 173–195, 2000.

[8] M. Khari, P. Kumar, D. Burgos, and R. G. Crespo, "Optimized test suites for automated testing using different optimization techniques," *Soft Comput.*, vol. 22, no. 24, pp. 8341–8352, Dec. 2018.

[9] C. A. C. Coello, G. B. Lamont, and D. A. V. Veldhuizen, *Evolutionary Algorithms for Solving Multi-Objective Problems*, 2nd ed. New York, NY, USA: Springer, 2007.

[10] A. Trivedi, D. Srinivasan, K. Sanyal, and A. Ghosh, "A survey of multiobjective evolutionary algorithms based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 21, no. 3, pp. 440–462, Jun. 2017.

[11] Q. Xu, Z. Xu, and T. Ma, "A survey of multiobjective evolutionary algorithms based on decomposition: Variants, challenges and future directions," *IEEE Access*, vol. 8, pp. 41588–41614, 2020, doi: 10.1109/ACCESS.2020.2973670.

[12] Z. Liang, R. Song, Q. Lin, Z. Du, J. Chen, Z. Ming, and J. Yu, "A double-module immune algorithm for multi-objective optimization problems," *Appl. Soft Comput.*, vol. 35, pp. 161–174, Oct. 2015.

[13] D. Wang and K. Lu, "Design optimization of hydraulic energy storage and conversion system for wave energy converters," *Protection Control Mod. Power Syst.*, vol. 3, no. 1, pp. 1–9, Dec. 2018.

[14] S. Arora and S. Singh, "An effective hybrid butterfly optimization algorithm with artificial bee colony for numerical optimization," *Int. J. Interact. Multimedia Artif. Intell.*, vol. 4, no. 4, pp. 14–21, 2017.

[15] Z. He, J. Zhou, L. Mo, H. Qin, X. Xiao, B. Jia, and C. Wang, "Multiobjective reservoir operation optimization using improved multiobjective dynamic programming based on reference lines," *IEEE Access*, vol. 7, pp. 103473–103484, 2019, doi: 10.1109/ACCESS.2019.2929196.

[16] M. Masood, M. M. Fouad, R. Kamal, I. Glesk, and I. U. Khan, "An improved particle swarm algorithm for multi-objectives based optimization in MPLS/GMPLS networks," *IEEE Access*, vol. 7, pp. 137147–137162, 2019, doi: 10.1109/ACCESS.2019.2934946.

[17] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

[18] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm for multi-objective optimization," in *Proc. Evol. Methods Design, Optimization Control Appl. Industrial Problems*, Athens, Greece, 2002, pp. 95–100.

[19] X. Zou, Y. Chen, M. Liu, and L. Kang, "A new evolutionary algorithm for solving many-objective optimization problems," *IEEE Trans. Syst., Man, Cybern.*, vol. 18, no. 4, pp. 577–601, 2014.

[20] Y. Hua, Y. Jin, and K. Hao, "A clustering-based adaptive evolutionary algorithm for multiobjective optimization with irregular Pareto fronts," *IEEE Trans. Cybern.*, vol. 49, no. 7, pp. 2758–2770, Jul. 2019.

[21] E. Zitzler and S. Kÿnzli, "Indicator-based selection in multi-objective search," in *Proc. Int. Conf. Parallel Problem Solving Nature-PPSN VIII*, 2004, pp. 832–842.

[22] N. Beume, B. Naujoks, and M. Emmerich, "SMS-EMOA: Multiobjective selection based on dominated hypervolume," *Eur. J. Oper. Res.*, vol. 181, no. 3, pp. 1653–1669, Sep. 2007.

[23] K. Li, S. Kwong, Q. Zhang, and K. Deb, "Interrelationship-based selection for decomposition multiobjective optimization," *IEEE Trans. Cybern.*, vol. 45, no. 10, pp. 2076–2088, Oct. 2015.

[24] W. Li, L. Wang, Q. Jiang, X. Hei, and B. Wang, "Multiobjective cloud particle optimization algorithm based on decomposition," *Algorithms*, vol. 8, no. 2, pp. 157–176, Apr. 2015.

[25] L. Wang, Q. Zhang, A. Zhou, M. Gong, and L. Jiao, "Constrained subproblems in a decomposition-based multiobjective evolutionary algorithm," *IEEE Trans. Evol. Comput.*, vol. 20, no. 3, pp. 475–480, Jun. 2016.

[26] C. Dai and X. Lei, "An improvement decomposition-based multi-objective evolutionary algorithm with uniform design," *Knowl.-Based Syst.*, vol. 125, pp. 108–115, Jun. 2017.

[27] J. Qiao, H. Zhou, C. Yang, and S. Yang, "A decomposition-based multiobjective evolutionary algorithm with angle-based adaptive penalty," *Appl. Soft Comput.*, vol. 74, pp. 190–205, Jan. 2019.

[28] H. Zhang, Y. Zhao, F. Wang, A. Zhang, P. Yang, and X. Shen, "A new evolutionary algorithm based on MOEA/D for portfolio optimization," in *Proc. 10th Int. Conf. Adv. Comput. Intell. (ICACI)*, Xiamen, China, Mar. 2018, pp. 831–836.

[29] Y. Zhang, D.-W. Gong, J.-Y. Sun, and B.-Y. Qu, "A decomposition-based archiving approach for multi-objective evolutionary optimization," *Inf. Sci.*, vols. 430–431, pp. 397–413, Mar. 2018.

[30] W. Lin, Q. Lin, Z. Zhu, J. Li, J. Chen, and Z. Ming, "Evolutionary search with multiple utopian reference points in decomposition-based multiobjective optimization," *Complexity*, vol. 2019, pp. 1–22, Apr. 2019.

[31] H. Li and Q. Zhang, "MOEA/D: A multi-objective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712–731, Dec. 2007.

[32] M. Wu, K. Li, S. Kwong, Q. Zhang, and J. Zhang, "Learning to decompose: A paradigm for decomposition-based multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 23, no. 3, pp. 376–390, Jun. 2019.

[33] F. Gu and Y.-M. Cheung, "Self-organizing map-based weight design for decomposition-based many-objective evolutionary algorithm," *IEEE Trans. Evol. Comput.*, vol. 22, no. 2, pp. 211–225, Apr. 2018.

[34] A. Camacho, G. Toscano, R. Landa, and H. Ishibuchi, "Indicator-based weight adaptation for solving many-objective optimization problems," in *Proc. 10th Int. Conf. Evol. Multi-Criterion Optim.*, East Lansing, MI, USA, Mar. 2019, pp. 216–228.

[35] X. Guo, X. Wang, and Z. Wei, "MOEA/D with adaptive weight vector design," in *Proc. 11th Int. Conf. Comput. Intell. Secur. (CIS)*, Dec. 2015, pp. 291–294.

[36] Q. Zhang, W. Liu, and H. Li, "The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances," in *Proc. IEEE Congr. Evol. Comput.*, May 2009, pp. 203–208.

[37] A. Zhou and Q. Zhang, "Are all the subproblems equally important? Resource allocation in decomposition-based multiobjective evolutionary algorithms," *IEEE Trans. Evol. Comput.*, vol. 20, no. 1, pp. 52–64, Feb. 2016.

[38] S. Jiang, S. Yang, Y. Wang, and X. Liu, "Scalarizing functions in decomposition-based multiobjective evolutionary algorithms," *IEEE Trans. Evol. Comput.*, vol. 22, no. 2, pp. 296–313, Apr. 2018.

[39] R. Wang, Z. Zhou, H. Ishibuchi, T. Liao, and T. Zhang, "Localized weighted sum method for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 22, no. 1, pp. 3–18, Feb. 2018.

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

[41] S. Das and P. N. Suganthan, "Differential evolution: A survey of the state-of-the-art," *IEEE Trans. Evol. Comput.*, vol. 15, no. 1, pp. 4–31, Feb. 2011.

[42] K. Li, A. Fialho, S. Kwong, and Q. Zhang, "Adaptive operator selection with bandits for a multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 18, no. 1, pp. 114–130, Feb. 2014.

[43] Q. Lin, Z. Liu, Q. Yan, Z. Du, C. A. C. Coello, Z. Liang, W. Wang, and J. Chen, "Adaptive composite operator selection and parameter control for multiobjective evolutionary algorithm," *Inf. Sci.*, vol. 339, pp. 332–352, Apr. 2016.

[44] Q. Zhang, A. Zhou, S. Zhao, P. N. Suganthan, W. Liu, and S. Tiwari, "Multiobjective optimization test instances for the CEC 2009 special session and competition," School Comput. Sci. Electron. Eng., Univ. Essex, Colchester, U.K., School Elect. Electron. Eng., Nanyang Technol. Univ., Singapore, Tech. Rep. CES-487, 2008.

[45] H. Li and Q. Zhang, "Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 13, no. 2, pp. 284–302, Apr. 2009.

[46] Z. Wang, Y.-S. Ong, J. Sun, A. Gupta, and Q. Zhang, "A generator for multiobjective test problems with difficult-to-approximate Pareto front boundaries," *IEEE Trans. Evol. Comput.*, vol. 23, no. 4, pp. 556–571, Aug. 2019.

[47] M. Li and J. Zheng, "Spread assessment for evolutionary multi-objective optimization," in *Proc. 5th Int. Conf. Evol. Multi-Criterion Optim.*, Nantes, France, 2009, pp. 216–230.

[48] L. While, P. Hingston, L. Barone, and S. Huband, "A faster algorithm for calculating hypervolume," *IEEE Trans. Evol. Comput.*, vol. 10, no. 1, pp. 29–38, Feb. 2006.

[49] S. García, D. Molina, M. Lozano, and F. Herrera, "A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: A case study on the CEC'2005 special session on real parameter optimization," *J. Heuristics*, vol. 15, no. 6, pp. 617–644, Dec. 2009.

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

[51] R. Wang, Q. Zhang, and T. Zhang, "Decomposition-based algorithms using Pareto adaptive scalarizing methods," *IEEE Trans. Evol. Comput.*, vol. 20, no. 6, pp. 821–837, Dec. 2016.

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

[53] Y. Zhang, D.-W. Gong, and Z. Ding, "A bare-bones multi-objective particle swarm optimization algorithm for environmental/economic dispatch," *Inf. Sci.*, vol. 192, pp. 213–227, Jun. 2012.

[54] D. Gong, B. Xu, Y. Zhang, Y. Guo, and S. Yang, "A similarity-based cooperative co-evolutionary algorithm for dynamic interval multiobjective optimization problems," *IEEE Trans. Evol. Comput.*, vol. 24, no. 1, pp. 142–156, Feb. 2020.

[55] Y. Hu, Y. Zhang, and D. Gong, "Multiobjective particle swarm optimization for feature selection with fuzzy cost," *IEEE Trans. Cybern.*, early access, Sep. 14, 2020, doi: 10.1109/TCYB.2020.3015756.

NAILI LUO received the B.S. degree in information and computing science from Zhaoqing University, Guangdong, China, in 2007, the M.S. degree in system theory from Guangdong Polytechnic Normal University, Guangdong, China, in 2011, and the Ph.D. degree in information and communication engineering from the College of Information Engineering, Shenzhen University, Guangdong, China.

She is currently a Postdoctoral Fellow with the College of Computer Science and Software Engineering, Shenzhen University. Her research interests include the theory and implementation of signal sampling and reconstruction, and the intelligent optimization algorithms, including the theory and design implementation of the evolutionary optimization algorithms.

WU LIN received the B.S. degree from the Hubei University of Technology, Wuhan, China, in 2017, and the M.S. degree from Shenzhen University, China, in 2020.

He is currently a Research Assistant with the College of Computer Science and Software Engineering, Shenzhen University. His research interests include evolutionary computation, multimodal multi-objective optimization, machine learning, and transfer optimization.

GENMIAO JIN received the B.S. degree from East China Jiaotong University, Nanchang, China, in 2012, and the M.S. degree from Shenzhen University, Shenzhen, China, in 2017.

He is currently a Software Engineer with China Merchants Bank, Shenzhen.

CHANGKUN JIANG (Member, IEEE) received the Ph.D. degree in information engineering from The Chinese University of Hong Kong (CUHK), in June 2017. He is currently an Assistant Professor with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China. His main research interest includes resource sharing and allocation for intelligent networked systems, by using tools from network games, optimization, and artificial intelligence.

JIANYONG CHEN received the Ph.D. degree from the City University of Hong Kong, China, in 2003.

He worked with ZTE Corporation as a Senior Engineer of the network technology, from 2003 to 2006. In 2006, he joined Shenzhen University, where he is currently a Professor with the College of Computer Science and Software Engineering. He is interested in artificial intelligence and information security. He has published more than 30 articles. He holds more than 30 patents in the fields of artificial intelligence and information security. He was the Vice-Chairman of the International Telecommunication Union-Telecommunication (ITU-T) SG17, from 2004 to 2012, and an Editor of three recommendations developed in ITU-T SG17.