Expert Systems With Applications 282 (2025) 127644

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Micro multi-objective genetic algorithm with information fitting strategy for low-power microprocessor

Hu Peng [a,b,*], Tian Fang [a], Jianpeng Xiong [a], Zhongtian Luo [a], Tao Liu [a,b], Zelin Wang [c]

[a] School of Computer and Big Data Science, Jiujiang University, Jiujiang, 332005, PR China
[b] Jiujiang Key Laboratory of Digital Technology, Jiujiang, 332005, PR China
[c] School of Information Science and Technology, Nantong University, Nantong, 226019, PR China

ARTICLE INFO

Keywords:
Micro multi-objective genetic algorithm
Information fitting strategy
Dual archives strategy
Adaptive mating selection

ABSTRACT

Micro multi-objective evolutionary algorithms (μMOEAs) are designed to address multi-objective optimization problems (MOPs), particularly in low-power microprocessor where computing resources are constrained. However, to compensate for the diversity loss resulting from using a micro population, existing optimization methods in numerous μMOEAs lead to diminished competitiveness over time due to the absence of targeted feedback on population states, hindering further performance improvement. To address this challenge, a micro multi-objective genetic algorithm with information fitting strategy for low-power microprocessor(μMOGAIF) is proposed, which utilizes an information fitting strategy to monitor the evolutionary status of the population and to facilitate method selection. The status information is collected at each iteration and fitted regularly, and the evaluation indicator is adjusted by the fitted evaluation results. In addition, adaptive mating selection is used in the construction of the mating pool to enhance the exploitation of solutions in probable regions. To enhance the adaptability of μMOGAIF, dual archives are established, one archive compensates the output using various strategies to pursue convergence or diversity, while the other provides the final output set. μMOGAIF is compared with five state-of-the-art MOEAs and five μMOEAs on the DTLZ, WFG, MaF, and ZDT benchmark test suites, and the experimental results demonstrate that μMOGAIF has outstanding performance. Furthermore, simulations based on low-power microprocessor have been conducted to verify the feasibility of μMOGAIF.

1. Introduction

Multiple objectives that require simultaneous optimization are commonly encountered in real-world problems, referred to as multi-objective optimization problems (MOPs) (Konak, Coit, & Smith, 2006; Peng, Xiong, Pi, Zhou and Wu, 2024; Sindhya, Miettinen, & Deb, 2013; Zhou et al., 2011) Various techniques have emerged to realize trade-offs between multiple conflicting sub-objectives in addressing MOPs. Goldberg et al. employed evolutionary algorithms as a technique in 1989 (Goldberg, 1989), known as multi-objective evolutionary algorithms(MOEAs), which was highly influential for subsequent research on solving MOPs (Censor, 1977). Various advanced MOEAs like NSGA-II (Deb, Pratap, Agarwal, & Meyarivan, 2002), MOEA/D (Zhang & Li, 2007), and μMOEA (Peng, Kong and Zhang, 2024) have been proposed for real-world applications, including smart home (Maurya & Nanda, 2023), smart agriculture (Kirimtat, Tasgetiren, Krejcar, Buyukdagli, & Maresova, 2024), micro robot (Mokhtari-Moghadam, Salhi, Yang, Nguyen, & Pourhejazy, 2025), and other domains (Gasmi, Soui, Barhoumi, & Abed, 2024).

As one of the most ubiquitous tools across diverse control domains, the low-power microprocessor (Quinn, Fairbanks, Tripp, Duran, & Lopez, 2014) represents a compact microcomputer system. Fabricated via large-scale integrated circuit technology, it consolidates a low-frequency central processing unit (CPU) alongside ancillary components onto a single silicon chip. Low-power microprocessors have the characteristics of being low-cost, consuming low energy, and having limited storage and computing resources. Their CPU frequencies fall significantly short of those in mainstream computers, with memory capacities typically under 1 MB. While traditional MOEAs are effective for solving MOPs, low-power microprocessors are indeed unable to sustain the memory and computing resources they consume (Peng, Kong et al., 2024; Peng, Xiao et al., 2022). For applications on low-power microprocessors, the purpose of MOEA is to find a micro-sized

* Corresponding author at: School of Computer and Big Data Science, Jiujiang University, Jiujiang, 332005, PR China.
E-mail addresses: penghu@jju.edu.cn (H. Peng), 3203665230@aliyun.com (T. Fang), XJP1972026766@aliyun.com (J. Xiong), Luo_Zhongtian@aliyun.com (Z. Luo), taoliu@jju.edu.cn (T. Liu), ntwzl@ntu.edu.cn (Z. Wang).

https://doi.org/10.1016/j.eswa.2025.127644
Received 14 October 2024; Received in revised form 29 March 2025; Accepted 7 April 2025
Available online 17 April 2025
0957-4174/© 2025 Elsevier Ltd. All rights reserved, including those for text and data mining, AI training, and similar technologies.

Pareto set efficiently rather than finding more optimal Pareto solutions. Consequently, micro population MOEAs (μMOEAs)emerge as the preferred choice for resource-constrained settings (Min, Hu, & Zhu, 2014).

While micro populations can address practical problems, they often result in considerable diversity loss within algorithms. As the number of objectives increases, maintaining diversity becomes more challenging (Zhou et al., 2011). Therefore, preventing the population from falling into local optima is the primary goal of most μMOEAs Alba and Dorronsoro (2005). In order to increase diversity, various methods are used in the proposed multi-objective population algorithm. The population restart mechanism directly enhances diversity (Cabrera & Coello, 2010; Liu, Han, & Jiang, 2012), yet introducing new randomly generated individuals without leveraging any learning history might impede convergence speed. The effective use of evolutionary operators, such as adjustment of the source of individuals (Nag, Pal, & Pal, 2015; Tiwari, Fadel, & Deb, 2011; Tiwari, Koch, Fadel, & Deb, 2008) or the utilization of the most efficient operator (Santiago, Dorronsoro, Fraire, & Ruiz, 2020), leads to a more uniform distribution of solutions near the Pareto front. In addition, in the conventional MOEAs, the large evolution population takes up many storage resources while providing a wealth of non-dominated solutions for evolution (Abdi, Asadpour, & Seyfari, 2023), which does not align with the requirements of low-power microprocessors.

A micro multi-objective genetic algorithm with an information fitting strategy for low-power microprocessors, termed μMOGAIF, is proposed to address these issues effectively. The information fitting strategy is used to collect information about the status of the population, and the construction of adaptive mating pools and the selection of methods suitable for the current archive are based on this. Through targeted feedback on the obtained information, μMOGAIF continuously converges towards the true Pareto front and achieves a uniform distribution simultaneously. The main contributions of the paper are as follows.

(1) An information fitting strategy is designed to supplement preference information in multiple processes involving real-time information collection and periodic information fitting.
(2) A dual archives strategy is used to maintain the comprehensive performance. The compensatory archive is introduced to improve distribution and convergence, while the essential archive aims at achieving the trade-off between convergence and diversity.
(3) To improve the exploitation rate of potential areas, the mating pool selects parents from two archives, respectively, which are adjusted based on dominance relationships and specific diversity ratios.

The structure of the paper is organized as follows. In Section 2, the multi-objective problems, micro population MOEAs, the related work, and the motivation are presented. Section 4 presents the process of the proposed μMOGAIF algorithm in detail. In Section 5, the performance of μMOGAIF and the other algorithms on 28 artificial problems was first tested, and further experiments were conducted on the low-power processor. Finally, we summarize the entire article and introduce future work in Section 6.

2. Preliminaries

This section overviews multi-objective optimization problems, micro multi-objective evolutionary algorithms, and recent work on μMOEAs.

2.1. Multi-objective optimization problems

A problem with M objectives to be optimized can be mathematically described as follows (Deb et al., 2002):

$$\begin{aligned} &Minimize \quad F(x) = \left(f_1(x), f_2(x), \ldots, f_m(x)\right) \in R^M \\ &s.t. \quad g(x) \le 0, h(x) = 0 \end{aligned} \tag{1}$$

where $x \in \left(x_1, x_2, \ldots, x_n\right) \in R^N$, R^N is the decision variable space, while R^M is the objective space. $F(x)$ represents a set of solutions in an M-dimensional space corresponding to solutions in the decision space. The M-dimensional space is also called the objective space.

For an M-dimensional minimization problem, there are two solutions, x_1 and x_2, from the decision space. It can be stated that x_1 dominates x_2 ($x_1 \prec x_2$) if the relationship between x_1 and x_2 is as follows:

$$\begin{cases} f_i\left(x_1\right) \le f_i\left(x_2\right) \forall i = 1,2,3,\ldots,m \\ f_i\left(x_1\right) < f_i\left(x_2\right) \exists i = 1,2,3,\ldots,m \end{cases} \tag{2}$$

A solution $x^* \in R^N$ is called the Pareto optimal solution as long as there is no solution $x \in R^N$ met $f(x)$ dominates $f(x^*)$. Finding a single solution that perfectly solves multiple problems is nearly impossible, so the usual approach is to get a set of solutions. A set of solutions is called the Pareto optimal set (PS) by comparing all solutions through various selection methods. The solutions in the objective space corresponding to PS are defined as the Pareto Front(PF) (Yu et al., 2022).

2.2. Micro multi-objective evolutionary algorithms

The main difference between micro multi-objective evolutionary algorithms(μMOEAs) and traditional MOEAs is the population size used in the evolutionary process (Pulido & Coello, 2003). The population size of traditional algorithms is generally between 30 and 200, while micro multi-objective algorithms generally do not exceed 15.

Goldberg's 1989 theoretical analysis (Goldberg, 1989) indicates that population convergence towards the front is unaffected by small population size or chromosome length. However, due to the small size of the evolutionary population, even starting from a randomly initialized population, the genes of individuals in the population will become very similar after a few generations, which is called nominal convergence (Coello & Pulido, 2001). Preventing premature local convergence necessitates robust population diversity maintenance methods, and is a primary focus of μMOEAs. This issue is also the key topic of our study. The background of micro population multi-objective algorithms and the work in diversity maintenance will be detailed in the next section.

2.3. Low-power microprocessor

Since the first microprocessor was designed in 1971 (Aspray, 1997), there has been a continuous release of microprocessors with improved performance, smaller sizes, and more affordable prices, widely known as low-power microprocessors. With low power consumption and easy scalability, low-power microprocessors are an ideal choice for IoT devices in areas such as smart home (Maurya & Nanda, 2023) and smart agriculture (Kirimtat et al., 2024). To achieve excellent low-power microprocessors, low-power process technology and the selection of device feature sizes play an important role (Gary, 1996). For contemporary low-power microprocessors, their memory capacity is insufficient to support the high cost of traditional multi-objective algorithms.

The fundamental principles of low-power microprocessors are similar to those of high-end computers. Still, they offer higher working stability, reduced power consumption, robust adaptability to the environment, a smaller physical footprint, and more integrated functions. Various low-power microprocessors have been designed and explicitly applied in different industries. Among commonly used low-power microprocessors, the 32-bit ESP32-WROOM is priced at only $5, and its core can achieve computing power of up to 600 MIPS, with a clock frequency ranging from 80 MHz to 240 MHz (Anshori et al., 2022). It is equipped with 448 KB of ROM, 520 KB of SRAM, 16 KB of RTC SRAM, and 40 MHz of internal oscillator. Texas Instruments designed the 16-bit MSP430 microcontrollers, most MCU memory developed based on which is less than 256 KB. Its clock frequency ranges from 4MHz to 25MHz, and it is priced at around $20 (Kashyap, Lakhani, Jain, & Agrawal, 2022). The TM4C1294NCPDT is a 32-bit Arm Cortex-M4F-based MCU with 1-MB flash, 256-KB RAM, and a clock frequency up to 120 MHz.

3. Related works

μMOEAs are a newly developed field due to their effectiveness in exploring and exploiting micro populations. Using a micro population leads to elite evolution but results in a lack of diversity and falling into local optima (Alba & Dorronsoro, 2005). As a population-based global search algorithm, genetic algorithms can avoid getting trapped in local optimal situations (Dutta, Sil, & Dutta, 2020). A series of μMOEAs is proposed based on the micro-GA framework. One of the earliest research achievements in applying micro populations to multi-objective optimization problems was proposed by Coello et al. in 2001 (Coello & Pulido, 2001). μGA adopts a population memory method with replaceable and non-replaceable portions, the proportion of which is adjusted by setting a fixed parameter. He also proposed μGA^2 (Pulido & Coello, 2003) in 2003 that used a parallel strategy of adapting the crossover operators and sharing the population following the comparison of results. Whether or not an operator is selected depends on evaluating the offspring it has generated. In 2008, Tiwari et al. (2008) proposed AMGA, which uses a large external archive size to obtain a large number of non-dominated solutions. Tiwari et al. (2011) proposed AMGA2 in 2011, which changes the mating pool composition and further improves the DE operator. G.P.Liu et al. Liu et al. (2012) proposed μMOGA in 2012, in response to the disadvantage of easily missing diversity in the micro populations, which uses an exploration operator to generate non-dominated solutions, and then adopts a population restart strategy. Nag et al. (2015) proposed an archive-based steady-state micro genetic algorithm (AMSiGA), in which an adaptive-size archive with upper and lower threshold settings is adopted, and the population chooses different operators and diversity maintenance methods based on the number of dimensions. Santiago et al. (2020) proposed μFAME in 2021, which promotes both diversity and accuracy of solutions through FIS, with the HV metric used to evaluate the offspring.

There are also micro-population algorithms based on evolutionary algorithm frameworks in addition to GA. Cabrera and Coello (2010) proposed μMOPSO in 2010, μMOPSO is the first combination of PSO and micro population, which maintains population diversity through population re-initialization and uses two archives to store the excellent solutions found during the search process and the final solution set separately. In 2022, Hu Peng et al. proposed a micro multi-strategy multi-objective ABC algorithm (μMMABC) (Peng, Wang et al., 2022), which uses a multi-strategy ABC optimizer and an adaptive updating mechanism to realize a proper trade-off between exploration and exploitation. It is worth noting that in the above algorithm, such as ASMiGA, the Pareto front solutions are directly used as the population, which means that its evolutionary population does not conform to the concept of a micro-sized population. In 2024, μMOEA was proposed, integrating the MOEA/D framework with a piecewise strategy and dynamic weight updates to adapt to resource-constrained environments.

In the work of those mentioned above μMOEAs, the most common diversity maintenance strategies are as follows.

(1) Variation operators: The variation of operators is used to address the premature convergence characteristic of micro population algorithms, endowing them with more randomness to resist it (Tiwari et al., 2011). μGA^2 (Pulido & Coello, 2003) dynamically selects operators to enable each population generation to generate the next generation using the best operator. ASMiGA (Nag et al., 2015) selects crossover operators based on the number of objects, variables, and constraints. The selection of crossover operators of μFAME (Santiago et al., 2020) is based on a FIS, which takes the evaluation results of a randomly selected descendant individual and the selection probability of the operator as inputs. The differential evolution (DE) operator (Babu & Jehan, 2003; Kukkonen & Lampinen, 2005; Storn & Price, 1997) has shown excellent results in AMGA2 and has begun to be actively applied in μ MOGAs, AMGA2 and AMSiGA successively proposed improved DE operators DE-2 and DE-3, both of which generate solutions of offspring as given by Eq. (3).

$$o_j = \begin{cases} (a_3)_j + F\left((a_1)_j - (a_2)_j\right) & if \quad r_j < CR \ or \ j = j_r \\ (p_i)_j & otherwise. \end{cases} \tag{3}$$

where j represents the jth variable. j_r refers to a random integer uniformly distributed between 1 and N, and N is the number of variables. r_j is another uniformly distributed random number between 0 and 1. p refers to the primary parent, and a_i refers to the auxiliary parents. F and CR are two tuning parameters.

(2) Crowding-distance (Deb et al., 2002): The metric is used as a fitness value to measure the diversity of the current population and is widely used in micro population multi-objective algorithms like AMGA, AMGA2, and μMOGA. Traditional crowding distance was proposed in NSGA-II; the fitness value assigned to an individual is the sum of the distances between it and its left and right neighbors on each objective. The relative distance between neighbors dramatically influences the results of this fitness assignment method. In the bounding boxes defined by the same neighbors, it is easy to encounter situations where individuals are assigned the same fitness despite being different.

(3) Hybrid methods: Integrating diverse techniques is a key factor in the performance of hybrid MOEAs. Appropriate combinations of different techniques may improve the behavior of the algorithm (Abdi & Feizi-Derakhshi, 2020). Hybrid MOEAs could be classified into two categories: hybridizing different search methods and hybridizing different approaches in different phases (Lu, Gao, Li, Zeng, & Zhou, 2018). μFAME uses a hybrid method to preserve population diversity, whereby disparate evolutionary operators are integrated and selected via a fuzzy inference system (Santiago et al., 2020). μMOSM is based on a novel multi-objective search manager (MOSM) framework, which hybridizes different evolutionary operators without increasing the time complexity (Abdi et al., 2023).

4. The proposed μMOGAIF

4.1. Motivation

Compared to traditional MOEAs, μMOEAs are more suitable for low-power microprocessors with limited storage and computing resources, but at the same time, maintaining the diversity of populations is more complicated. If the diversity of the population is not maintained promptly, the population will soon stagnate at a distance from the true PF. The use of inappropriate strategies can also disrupt population evolution, causing slower convergence or uneven distribution. Accurately assessing the evolutionary state of the population and providing timely feedback to balance convergence and diversity requires the algorithm to possess strong adaptive adjustment capabilities.

It is usually implemented by the re-initialization method, improving mating selection techniques, and selecting appropriate operators in the existing μMOEAs to improve the diversity. However, falling into local optima has not been fully addressed, and the algorithm becomes

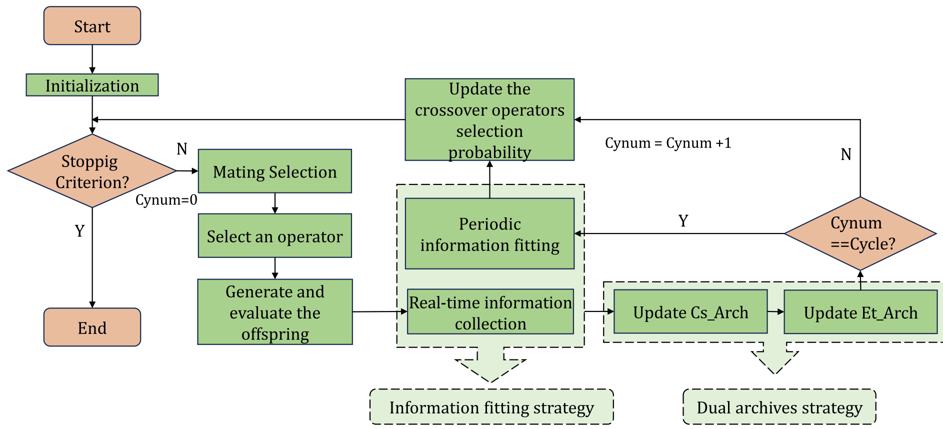


Fig. 1. The flow-chart of μMOGAIF. The information fitting strategy is the critical part, consisting of real-time information collection and periodic information fitting. Cs_Arch and Et_Arch are serialized in the dual archives strategy, outputting an approximate PF in each iteration. Cynum is a variable that records the number of iterations in a cycle.

less competitive over time. Hybrid methods combine different techniques and have received attention due to their good performance. Compared to the selection criteria based on the number of objectives in ASMiGA (Nag et al., 2015), μFAME hybridizes different operators and selects based on the current status of the population, maintaining diversity through the adaptability of operator selection (Santiago et al., 2020). In this paper, the information fitting strategy is used to gather and fit the information of the population to assist the adaptive processes involving the mating selection, the operator selection, and the truncation in archives.

In addition, dual archives are established to pursue different goals separately to balance exploration and exploitation. Compared to using an external archive in previous μMOGAs, μMOGAIF has added a compensatory archive to assist the population in exploring more probable search regions. By simultaneously combining the evaluation results obtained from the information fitting strategy, the selection strategy in the archive is adjusted.

4.2. The framework of μMOGAIF

The goal of μMOEAs is to obtain a set of solutions with uniform distribution over the PF, but using a micro population increases the difficulty of maintaining population diversity, which hinders the convergence of the population from approaching the true PF. Most existing μMOEAs use diversity maintenance strategies such as variation operators during the evolution process, which is prone to losing competitiveness over time (Nag et al., 2015). In this section, a micro multi-objective genetic algorithm with information fitting strategy(μMOGAIF) is proposed for low-power microprocessor with low-cost requirements. The overall process of the algorithm is shown in Fig. 1.

The information fitting strategy is the key to μMOGAIF, while adaptive mating selection and dual archives strategy assist in further improving the adaptive ability of the population. The pseudo-code is present in Algorithm 1. The population, archives, and parameters are initialized at first on lines 1 to 3. In μMOGAIF, mating selection, reproduction, and environmental selection are the main processes at each iteration. To prevent getting stuck in local optima, two type parent populations $Parents\,E$ and $Parents\,C$ are selected during the adaptive mating selection to increase the disturbance between vectors in line 5. During the reproduction in lines 6 to 7, an operator is chosen to carry out the crossover operation according to the selection probabilities. In line 8, a real-time information collection method is adopted to evaluate the newly generated offspring population and update the selection probability of operators. To output the excellent approximate front of the current iteration, a dual archives strategy comprises the updating processes of two external archives, the compensatory archive and the essential archive. In line 10, the essential archive is considered as the output. Once the value of the iteration counter has reached the predefined maximum number, μMOGAIF starts activating the second part of the information fitting strategy. In line 12, the periodic information fitting method assists the variables such as the iteration counter to restart.

Algorithm 1 The framework of μMOGAIF

Input: Number of population NP, archive size N
Output: Essential Archive Et_Arch
1: Randomly initializing population Pop, compensatory Archive Cs_Arch and Essential Archive Et_Arch
2: Assign all crossover operators with the same selection probability $OpProb$
3: assessed value of offspring $Scorepercent$, iteration number $Cynum$, optimal offspring evaluation value $MaxScore = 0$
4: **while** Stopping criterion not reached **do**
5: $Parents\,C$, $Parents\,E$ = Mating Selection(Cs_Arch, Et_Arch, $Scorepercent$)
6: Select the operator for evolution
7: Generate offspring
8: ($Scorepercent$, $Table$) = Real-time information collection(offspring, Et_Arch)
9: Cs_Arch = Update Compensatory_Arch(Et_Arch, offspring, $Scorepercent$, $MaxScore$, N)
10: Et_Arch = Update Essential_Arch(Cs_Arch, Et_Arch, N)
11: $Cynum = Cynum+1$
12: **if** $Cynum == Cycle$ **then**
12: $OpProb$, $MaxScore$ = Periodic information fitting($OpProb$, $Table$)
13: **end if**
14: **end while**

4.3. Information fitting strategy

To enable the population to remain competitive over the long run, an information fitting strategy is designed based on fuzzy logic to supplement preference information in choosing operators, mating selection, and updating archives. An information fitting strategy is the operation of gathering data and fitting the collected data via membership functions. The strategy involves two processes: real-time information collection and periodic information fitting. The information mentioned in this strategy is based on the judgment results of dominance relationships, which digitize the evolutionary state of the population.

4.3.1. Real-time information collection

Due to the continuous accumulation of selection probabilities without any re-initialization, and the constantly changing population situation, the operator selected based on historical cumulative data is not the most suitable option for the current generation. To improve the efficiency of μMOGAIF to converge to the global optimal PF, operators promoting population convergence are prioritized for selection. The evaluation criteria for operators depend on the contribution of the newly generated solutions in exploration relative to the individuals in the archive. Each solution within the newly generated offspring population is compared with the archive Et_Arch. The evaluation result is documented as a variable named $Scorepercent$.

The Pareto dominant relation between the newly generated solutions and the solutions in the archive can be classified as dominate, dominated, and non-dominated. A solution whose offspring individual dominate number greater than being dominated can be counted in $Scorepercent$. Thus, $Scorepercent$ denotes the proportion of these individuals in the current offspring.

Algorithm 2 Real-time information collection.

Input: Et_Arch, $offspring$, $MaxScore$, operator selection probability $OpProb$
Output: $Scorepercent$, partial offspring $Partone$, operator probability recording table $Table$

$Partone = \varnothing$
for $i = 1 : |offspring|$ **do**
  for $j = 1 : |Et\_Arch|$ **do**
    if $offspring[i]$ dominates $Et\_Arch[j]$ **then**
      Dominates = Dominates + 1
    else
      Dominated = Dominated + 1
    end if
  end for
  if Dominates > Dominated **then**
    $Score = Score + 1$
    $Partone = Partone \cup offspring[i]$
  end if
end for
$Scorepercent = Score$ / length($offspring$)
$Table[index][Cynum] = Scorepercent$
if $Scorepercent > 0.5 \times MaxScore$ **then**
  $OpProb$(index) = $OpProb$(index) + 1/ $Cycle$
else
  Randomly select an operator for probability update
end if

Within a single cycle, in addition to the different basis for probability changes, operator selection is also distinct from roulette wheel selection in μFAME, the tournament selection mechanism is applied to choose the operator to be utilized according to their current selection probabilities(OpProbs). The operators we consider are about crossover, and the operator pool is composed of GA, SBX and DE-3. OpProbs undergoes a real-time update upon the completion of the evaluation of the newly generated offspring. If the evaluation result $Scorepercent$ is more significant than 0.5* $MaxScore$, the selection probability of the chosen operator will be increased by $1/Cycle$. Conversely, if this criterion is not met, the reward value is added to the selection probability of any one operator. $MaxScore$ refers to the maximum value of $Scorepercent$ during the last cycle, which also demonstrates the ability of the operators in this stage. Incorporating $MaxScore$ as preference information into the judgment criteria is more appropriate than fixed or random numbers. More strategy details are shown in Algorithm 2.

4.3.2. Periodic information fitting

At the end of each cycle, the sets of evaluation results from the offspring populations are processed as information. The performance of each operator during the period is quantified as a numerical value between 0 and 1 using the membership function. In this paper, the generalized bell-shaped membership function with periodic adaptive shape adjustment is used to map the inputs and subsequently fit the outputs (Mahto, Saha, & Das, 2019). The shape of the generalized bell-shaped membership function is a bell-shaped curve, which has a steepness of the transition from 0 to 1 (Behnamian & Ghomi, 2014). The function is defined as Eq. (4).

$$\mu(x; a, b, c) = \frac{1}{1 + \left|\frac{x-c}{a}\right|^{2b}} \tag{4}$$

The shape parameters of the function a and b control the slopes at the cross points, where a is maintained at 0.2. The center point c determines the center of the membership function. The effects of changing the parameters of the membership function are shown in Fig. 2.

Considering that there are significant differences in the characteristics of offspring evaluation results across different evolutionary stages with long inter-cycle intervals, adaptively generating membership functions tailored to the current cycle via parameter adjustment is necessary. The degree of slope is controlled by the parameter b defined in Eq. (5).

$$b = \max(Scorepercent)/2 \tag{5}$$

The parameter c is determined by Eq. (6), where $Scorepercent$ and $Cycle$ have been defined in Section 4.3.1.

$$c = \frac{1}{Cycle} \sum_{i \le Cycle} Scorepercent(i) \tag{6}$$

This means that the value of b is half of the maximum evaluation value of the offspring in the previous period, and the central value of the function is the average evaluation value of the previous period. If the performance of the operator in the current cycle is similar to that in the previous cycle, then a restart probability of approximately 1 can be obtained. While periodically restarting operator selection probability can mitigate the influence of prolonged historical accumulation, operators may still fail to accumulate sufficient probability within a cycle owing to infrequent usage, despite showing strong current performance. For the fuzzy result of the member function mapping the input ($Table$) from clear space to fuzzy space, the centroid defuzzification method is applied to calculate a single clear output value. The centroid defuzzification method returns the center of gravity of the fuzzy set along the x-axis, the centroid is computed using the following Eq. (7):

$$Centroid(x) = \frac{\sum_i \mu(x_i) x_i}{\sum_i \mu(x_i)} \tag{7}$$

where $\mu(x_i)$ is the membership value for point x_i. The value of $Centroid$ will become the operator selection probability for the new cycle. The pseudo-code of periodic information fitting is shown in Algorithm 3.

Algorithm 3 Periodic information fitting.

Input: $Table$, $OpProb$, $MaxScore$
Output: $OpProb$, $MaxScore$

Calculate the parameters b and c by Eq.(5) and Eq.(6)
$MaxScore = c$
Adjust the shape of the membership function by Eq.(4) and parameters
Map crisp inputs $Table$ into fuzzy space by Eq.(4)
Calculate the centroid of each fuzzy set by Eq.(7)
OpProb = Centroid

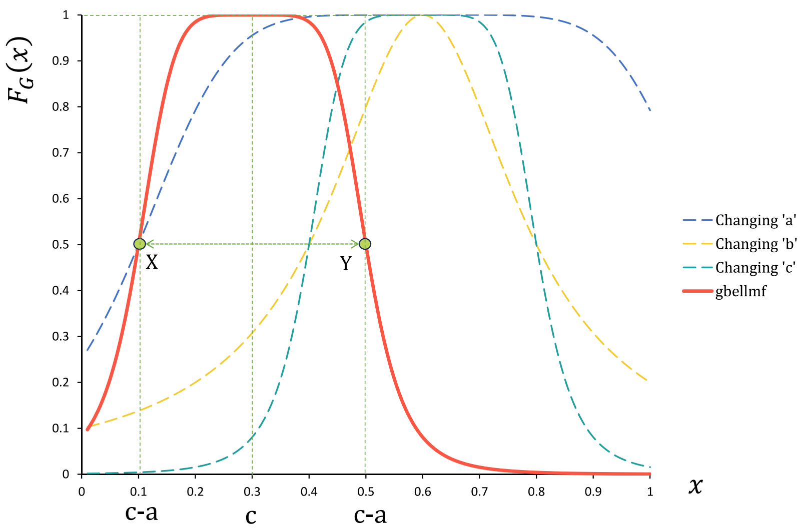


Fig. 2. The generalized bell-shaped membership function. The shape adjustment depends on three parameters a, b and c, a is the half width; b (together with a) controls the slopes at the crossover points; and c determines the center of the corresponding membership function. The effect of adjusting parameters is as above (Changing 'a', Changing 'b', Changing 'c') .

4.4. Dual archives strategy

Compared to using a single external archive in μMOGA algorithms, μMOGAIF uses the two archive techniques to set a compensatory archive and an essential archive (Hu et al., 2022). μMOGAIF uses two archival techniques, which include the establishment of a compensatory archive and an essential archive. Dual archives maintain a fixed size by using a dual archives strategy. The essential archive serves as the final output set, while the compensatory archive assists in balancing convergence and diversity.

4.4.1. Compensatory archive

The compensatory archive (Cs_Arch) is designed to provide solutions for the final output set and serve as the source of the primary parents in the mating pool. As stated in Section 4.5, offspring will be centered around the individuals used to establish the mating pool in Cs_Arch. Thus, Cs_Arch employs two different selection methods (Peng et al., 2023) to maintain a fixed size and realize a trade-off between exploration and exploitation. The choice of strategy is based on the information obtained from the information fitting strategy, which is described in detail in Section 4.3.

To encourage the population to converge to the true PF, the indicator $I_{\epsilon+}$ in IBEA (Zitzler & Künzli, 2004) is used as one of the selection strategies. This indicator represents the minimum distance required for one solution to dominate another solution in the objective space, as defined by Eq. (8).

$$I_{\epsilon+}(A,B) = min_{\epsilon}\left\{\forall x^2 \in B, \exists x^1 \in A : f_i\left(x^1\right) - \epsilon \le f_i\left(x^2\right) for i \in \{1,\dots,n\}\right\} \tag{8}$$

The fitness values assigned to individuals are obtained using Eq. (9).

$$F(x_1) = \sum\nolimits_{x_2 \in P\setminus\{x_1\}}^{-e^{-I_{\epsilon+}(x_2,x_1)/0.05}} \tag{9}$$

When the archive Cs_Arch is in an overflow state after adding offspring, the additional solutions are deleted one by one, and the fitness value of each solution is updated simultaneously. The individual with the lowest fitness value will be deleted first, and the final solution set is the updated Cs_Arch with a fixed number.

On the contrary, another updating strategy of Cs_Arch aims to maintain population diversity. In Section 4.3, individuals in newly generated offspring with strong convergence are stored in a solution set named $Partone$. $Partone$ will first be retained to provide strong and effective convergence momentum for the PF. For vacant positions in Cs_Arch, individuals with high crowding distance values are given priority for filling the archive. The crowding distance value is given by Eq. (10)

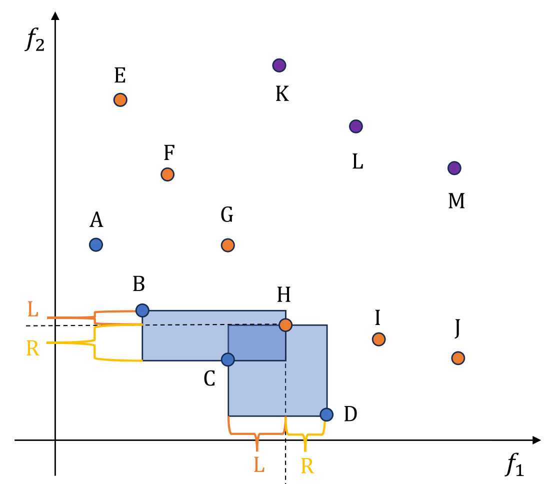


Fig. 3. Crowding-distance assignment. The blue points refer to individuals in the first level, while the orange points refer to individuals in the next level. L and R refer to the Euclidean distances between individuals and their left and right neighbors.

$$CD_i = \sum_{j=1}^{k} (k - j + 1)\, d_{i,j} \tag{10}$$

As shown in Fig. 3, in NSGA-II, the individuals are divided into multiple levels. The first level includes A, B, C, D, and the second level comprises E through J in the diagram. For individual H, the left and right neighbors are G and I, while the distance between G and H is shorter than between H and any of its neighbors. When the bounding boxes of individuals are similar or consistent, the fitness values obtained are also identical. In such cases, the product of the distances between the individual and its left and right neighbors is used instead of the sum. Updating the compensatory archive details are described in Algorithm 4.

Algorithm 4 Update compensatory archive

Input: Et_Arch, offspring, N, $Scorepercent$, $MaxScore$
Output: Cs_Arch

```
while |Cs_Arch| > N do
    if Scorepercent < 0.5 × MaxScore then
        Delete the individual with the minimal CsF(i)
        Update the fitness CsF(i) = CsF(i) + e^{-I_{ε+}(i,j)/0.05}
    else
        Find offspring individuals Cs_part_1 that can dominate individuals in the Et_Arch after evaluation.
        Select individuals Cs_part_2 to fill the remaining vacant spaces based on the crowding distance.
        Cs_Arch=Cs_part_1 ∪ Cs_part_2
    end if
end while
```

Algorithm 5 Update essential archive

Input: Et_Arch, Cs_Arch, N, NP
Output: Et_Arch

```
The number of division div = NP
Calculate the grid location of each individual and the reference point
while |Et_Arch| > N do
    Calculate the grid ranking GR(i) of each individual
    Assign corresponding proportional values G(i) to individuals at the same level
    Calculate raw fitness R(i) based on dominance
    EtF(i) = G(i) + R(i)
    Sort ascending current population by Eq.(11) and discard redundant solutions
end while
```

4.4.2. Essential archive

Compared to the compensatory archive (Cs_Arch) with a preference selection tendency mentioned earlier, the essential archive (Et_Arch) serves as the storage archive for the output set. It needs to achieve a balance between convergence and diversity. The fitness assignment in SPEA2 is adopted in Et_Arch, and the fitness value can be formulated as Eq. (11).

$$EtF(i) = R(i) + G(i) \tag{11}$$

where the raw fitness, denoted as $R(i)$, represents the convergence assessment, and the grid ranking, denoted as $G(i)$, represents the diversity.

In calculating raw fitness $R(i)$, each individual i is assigned a strength value, $S(i)$, representing the number of solutions it dominates. The raw fitness of an individual i is determined by the sum of the strength value S of the individuals that dominate i. When the value of R is 0, it indicates that the individual is a non-dominated solution.

Grid raking of an individual refers to the total sum of grid coordinates for each objective. In contrast to the traditional method of using grid ranking as a convergence indicator, the proportion of each ranking is assigned as a fitness value G to individuals at their ranking in this paper. When assigning fitness value $G(i)$, the number of individuals at each level is considered to ensure a uniform distribution of solutions across multiple rankings.

Grid raking reflects the comprehensive performance of individuals across all dimensions. Even when there is a significant difference in specific dimensions and a large Euclidean distance between individuals, their contribution to the final output set may remain the same. To further improve population distribution, a reference point is established for the grid coordinates of the population. The reference point coordinates are calculated as the mean coordinates of the population on each dimension. The final grid coordinate of an individual will be the difference between the original grid coordinate and the reference point coordinate. The fitness value G referenced in the individual selection process is based on the approximate PF of the entire population and is not solely influenced by extreme individuals. The process for updating the essential archive is described in Algorithm 5.

4.5. Adaptive mating selection

To increase searching in more probable search regions, many μMOGAs establish the mating pool consisting of primary parents and auxiliary parents (Nag et al., 2015; Tiwari et al., 2011, 2008) to enhance the disturbance between vectors. In this algorithm, two types of parents are selected from Cs_Arch and Et_Arch respectively, named $Parents\,E$ and $Parents\,C$.

As the primary parents, there is a high probability that the offspring inherits the characteristic of $Parents\,E$. Hence, the individuals in $Parents\,E$ are chosen from the compensatory archive Cs_Arch based on dominance relationships. The archive Cs_Arch is constantly updated with different methods based on the offspring evaluation result $Scorepercent$. Compared with the output set Et_Arch, Cs_Arch has more potential in exploration during the evolutionary process and is a good choice for guiding the generation of new populations. A detailed description of Cs_Arch is provided in Section 4.4.1. For example, when the DE operator is chosen, $Parents\,E$ will be p_i in Eq. (3).

Auxiliary parents as another part of the mating pool, introducing randomness to the evolution of the population. This helps prevent premature convergence while maintaining invariance to affine transformations in the search space. Thus, as the approximate PF of μMOGAIF, Et_Arch provides individuals for $Parents\,C$ to participate in mating based on randomness and crowding distance. $Parents\,C$ consists of Et_1 (randomness) and Et_2 (crowding-distance) selected through different methods with constantly changing probabilities p_1 and p_2, where $p_1 = Scorepercent$, $p_2 = 1 - Scorepercent$. $Scorepercent$ determines the ratio of two components. As the population evolves, the proportion of individuals in newly generated offspring with strong dominance decreases. The value of $Scorepercent$ decreases, and the auxiliary parents are more based on diversity rather than randomness. $Parents\,C$ used in DE operator will be a_1, a_2 and a_3 in Eq. (3).

Algorithm 6 Adaptive mating selection.

Input: Cs_Arch, Et_Arch, $Scorepercent$, NP
Output: primary parents $Parent\,C$, auxiliary parents $Parent\,E$

```
Randomly select population Cs_1 and Cs_2 of size N/2 from Parent C

Record Pareto relationships between individuals in Cs_1 and Cs_2 as Dominance
Parent C = Parent C ∪ i for i in Cs_1 and dominates any individual in Cs_2
Parent C = Parent C ∪ j for j in Cs_2 and dominates any individual in Cs_1
p = rand(1,NP)
Et_1_len = length (find (p < Scorepercent))
Et_2_len = N - p_1
Randomly select p_1_len individuals as Et_1 from Cs_Arch
Select p_2_len individuals as Et_2 from Et_Arch based on crowding-distance
Parent E = Et_1 ∪ Et_2
```

5. Experimental results and analysis

5.1. Comparison algorithms

The performance of the proposed μMOGAIF is evaluated by comparing it with twelve state-of-the-art algorithms, which are as follows.

- AMGA (Tiwari et al., 2008): A micro genetic algorithm with improved formulation for diversity preservation techniques.
- AMGA2 (Tiwari et al., 2011): A micro genetic algorithm employs a new kind of selection strategy that attempts to reduce the probability of exploring less desirable search regions.
- Micro-MOPSO (Cabrera & Coello, 2010): A multi-objective evolutionary algorithm based on particle swarm optimization with a very small population size.
- μMOGA (Liu et al., 2012): A micro genetic algorithm adopts a restart strategy and an exploratory operator based on the degree of convergence of the evolutionary population.
- μMMABC (Peng, Wang et al., 2022): A new micro multi-strategy multi-objective ABC algorithm, which employs multi-strategy ABC optimizer and the adaptive updating mechanism, works together to achieve a trade-off between the exploration and exploitation.
- μMOEA (Peng, Kong et al., 2024): A micro MOEA with a piecewise strategy for industrial optimization in embedded processors introduces an improved piecewise strategy based on the MOEA/D framework.
- NSGA-III (Deb & Jain, 2014): A genetic algorithm using reference points based on the NSGA-II framework that emphasizes non-dominated population members yet close to a set of supplied reference points.
- Two_Arch2 (Wang, Jiao, & Yao, 2015): A hybrid MOEA based on two archives, which focus on convergence and diversity separately and combines the advantages of indicator and Pareto-based MOEAs.
- SPEA2+SDE (Li, Yang, & Liu, 2014) : To modify the diversity maintenance mechanism, SPEA2+SDE propose a shift-based density estimation (SDE) strategy.
- GrEA (Yang, Li, Liu, & Zheng, 2013): A grid-based evolutionary algorithm using three criteria, grid ranking, grid crowding distance, and grid coordinate point distance in mating and selection.
- IBEA (Zitzler & Künzli, 2004): A general indicator-based evolutionary algorithm by defining the optimization goal in terms of a binary performance measure (indicator) and using the measure in the selection process.
- MOCPSO (Zhang, Li, Hong, & Zhou, 2023): A Multi-Objective Cooperative Particle Swarm Optimization Algorithm with Dual Search Strategies.

5.2. Benchmark problems

To ensure fair and comprehensive experimentation, four test suites are used, covering different types of problems.

- DTLZ (Deb, Thiele, Laumanns, & Zitzler, 2005) has concave, multi-modal, biased, scaled, and disconnected characteristics.
- WFG (Huband, Hingston, Barone, & While, 2006) has characteristics such as nonseparable, deceptive, truly degenerate, mixed shape Pareto front, scalable in the number of position-related parameters, with dependencies between position- and distance-related parameters.
- ZDT (Zitzler, Deb, & Thiele, 2000) has characteristics such as convex, non-convex, and non-contiguous convex Pareto fronts and containing many local Pareto-optimal fronts.

Table 1
Parameter settings for the comparison algorithms.

Algorithm	Parameter setting
AMGA	–
AMGA2	–
μMOGA	Convergence criterion = 3
μMOPSO	p_r = 2, g_r = 100, AAB = 40, archive_size(FAB) = 20, mutation_rate = 0.1, pps = 0.2, C_1 = 1.8, C_2 = 1.8
μMMABC	rate_evol = 0.8, threshold = 50
μMOEA	F = 0.5, CR = 0.5, T=5, τ = 4, ξ_* = 1.00E−04, δ = 0.06, γ = 0.4
NSGA-III	–
Two_Arch2	–
GrEA	Div = 45,15 for bi-objective and tri-objective MOPs
IBEA	–
SPEA2+SDE	–
MOCPSO	–
μMOGAIF	Cycle = 10

- MaF (Gee, Tan, & Abbass, 2017) is a benchmark test suite for evolutionary many-objective optimization, which covers a good representation of various real-world scenarios, such as being multi-modal, disconnected, degenerate, and/or nonseparable, and having an irregular Pareto front shape, a complex Pareto set or a large number of decision variables.

5.3. Performance metrics

In this study, two widely accepted metrics are selected to evaluate the performance of each MOEA.

- Inverted Generational Distance (IGD) (Zitzler, Thiele, Laumanns, Fonseca, & Da Fonseca, 2003) is a common testing metric to evaluate the comprehensive performance of algorithms. The smaller the IGD value, the better the performance of the algorithm. The IGD value is given by the Eq. (12):

$$IGD\left(P, P^*\right) = \frac{\sum_{x \in p^*} \min_{y \in p} d\left(x, y\right)}{\left|P^*\right|} \tag{12}$$

where P is an approximate solution set of the PF obtained by an algorithm, P^* is a set of uniformly distributed sampling points of the real PF, $d(x, y)$ is the Euclidean distance between the individual x in P^* and the individual y in P.
- Spread (Nebro et al., 2008) is an extension of metric Spacing, as the latter only applies to two-dimensional problems. The metric computes the distance from a given point to its nearest neighbor, which indicates that the algorithms with larger Spread values are desirable. If the solutions in the output set are well distributed and include those extreme solutions, $\triangle = 0$. The Spread value is given by the Eq. (13):

$$\triangle = \frac{\sum_{i=1}^{m} d\left(e_i, S\right) + \sum_{X \in S} \left|d\left(X, S\right) - \bar{d}\right|}{\sum_{i=1}^{m} d\left(e_i, S\right) + |S| * \bar{d}} \tag{13}$$

$$d\left(X, S\right) = \min_{Y \in S, Y \neq X} \|F\left(X\right) - F\left(Y\right)\|^2 \tag{14}$$

$$\bar{d} = \frac{1}{|S^*|} \sum_{X \in S^*} d\left(X, S\right) \tag{15}$$

where S is a set of solutions, S^* is the set of Pareto optimal solutions, $e_1, \ldots, e_m$ are m extreme solutions in S^*, m is the number of objectives.

5.4. Parameter settings

In the experiment, the population size of the traditional algorithms is set to 20. In contrast, the population size of the micro population algorithms is set to 4*M, where M represents the dimension of the objective space. Besides, the archives in the comparison algorithms are all set to size 20. In the crossover and mutation parameter settings, the distribution index of the crossover operator SBX η_c is set to 20, and the probability p_r is set to 1.0. The DE crossover operator assigns values of 0.5 and 1.0 to the F and CR parameters. The distribution index of the polynomial mutation $\eta_m = 20$, and the mutation probability is represented as $p_m = 1/\text{number of variables}$. The parameter settings specific to each algorithm are detailed in Table 1, and are consistent with the literature on their algorithms. All experiments were conducted using PlatEMO (Version 2.9) (Tian, Cheng, Zhang, & Jin, 2017) and MATLAB 2018b. Each algorithm was run independently 30 times on every test instance, with 10 000 evaluations. The operating system is Windows 10, running on a 3.60 GHz CPU with 8 GB of RAM. The source code of μMOGAIF can be downloaded from Hu Peng's homepage: https://whuph.github.io/index.html.

5.5. Experimental results

5.5.1. Compared with traditional MOEAs

In this section, μMOGAIF is compared with five traditional MOEAs on 28 benchmark problems, including DTLZ, WFG, ZDT, and MaF. The results for evaluation metrics are presented in Tables 2 and 3, with the best results highlighted in blue. According to the Wilcoxon rank sum test at the 95% confidence level, '+', '−', and '≈' are used as symbols to display the comparison results when μMOGAIF is worse than, better than, or similar to the compared algorithms.

Table 2 presents the results for the IGD evaluation metric. On the DTLZ benchmark, μMOGAIF achieves the best value on DTLZ5 and DTLZ6. For the MaF benchmark, μMOGAIF achieved the best value except MaF3, MaF4, and MaF5, where μMOGAIF failed to converge to the true PF. For the WFG benchmark, μMOGAIF achieved the best value on WFG1, WFG4, WFG5, WFG7 and WFG9. On WFG2 and WFG6, although μMOGAIF did not achieve the optimal value, there was no significant difference compared to NSGA-III. For the ZDT benchmark, μMOGAIF has achieved excellent results except ZDT4. Overall, μMOGAIF obtained 13 best IGD mean values out of 28 test functions in terms of IGD. According to Fig. 5, μMOGAIF achieved the best ranking results of 3.32 by the Friedman test.

As shown in Table 3, μMOGAIF obtained 22 best Spread mean values. MOCPSO outperformed μMOGAIF on two tests, while NSGA-III, GrEA, and IBEA have no significant surpassing result compared to μMOGAIF. Fig. 6 shows the distribution of the final non-dominated solutions obtained by μMOGAIF on four instances. On the problem where μMOGAIF did not achieve the optimal value, the variance value of the result is larger compared to other issues. The statistical results of the Friedman test are performed in Fig. 5. Each row represents the ranking of one algorithm, where the orange denotes the ranking values on metric IGD, and the blue refers to the ranking values on metric Spread. μMOGAIF achieved ranking results of 3.32 and 1.93, both of which are the lowest values on the evaluation indicators. Regarding overall performance, Two_Arch2 is ranked second, and its strong performance on the diversity indicator is a significant contributing factor. Although NSGA-III ranks second in the IGD indicator, it is significantly weaker than Two_Arch2 in the Spread indicator, thus ranking third overall. Fig. 4 shows the distribution of the final non-dominated solutions obtained by μMOGAIF in several representative problems.

5.5.2. Compared with micro MOEAs

This section provides a detailed comparison experiment of micro population multi-objective algorithms. The six micro MOEAs compared include three μMOGAs, one μMOPSO, one μMOEAD and one μMOABC algorithm. Tables 4 and 5 show the average IGD and Spread obtained for μMOGAIF and five micro MOEAs. μMOGAIF achieved 17 best IGD values and 22 best Spread values out of 28 test instances. In Table 4, μMMABC is only inferior to μMOGAIF and performs best on 10 test instances, with five significantly exceeding μMOGAIF. Like the results in Table 3, μMOGAIF also showed superior performance on metric Spread in Table 5. In Fig. 5, AMGA, μMOGA, and μMOPSO have similar rankings in both IGD and Spread indicators, while AMGA2 and μMMABC have a significant difference in two rankings. μMMABC only achieved a better average Spread value than μMOPSO and μMOEA, but ranked second overall. Algorithm AMGA2, in contrast, ranked second on Spread but obtained a weaker overall ranking. Only μMOGAIF achieved a fairly good ranking on both indicators, representing that μMOGAIF has a strong ability to balance the convergence and diversity of optimizing MOPs with the micro population.

Average IGD values and Spread values of the seven algorithms over generations on DTLZ2 with three objectives are shown in Fig. 6. At the beginning of population evolution, μMOGAIF converges to the PF in the first few iterations. In the following iterations, μMOGAIF maintained its advantages and consistently outperformed other algorithms. As in Fig. 6(a), μMOGAIF does not fall into local optima too early like the other five algorithms and has a stronger ability to explore the true PF. In Fig. 6(b), except for μMOGAIF and μMOGAIF, the curves of the other four algorithms fluctuated greatly. μMOGAIF demonstrates its robustness in maintaining diversity.

5.5.3. Performances on many-objectives problems

To test the performance of the algorithms on higher dimensions, μMOGAIF, and the other 10 algorithms were compared on 24 5-objective problems, which include MaF benchmark and WFG benchmark (Wang & Ma, 2024). Tables 6 and 7 presents the comparisons between μMOGAIF and five traditional MOEAs. μMOGAIF obtained 8 best IGD and 20 best Spread values out of 24 test instances. The experimental results between μMOGAIF and five micro MOEAs are shown in Tables 8 and 9. μMOGAIF obtained 16 best IGD and 22 best Spread values out of 24 test instances.

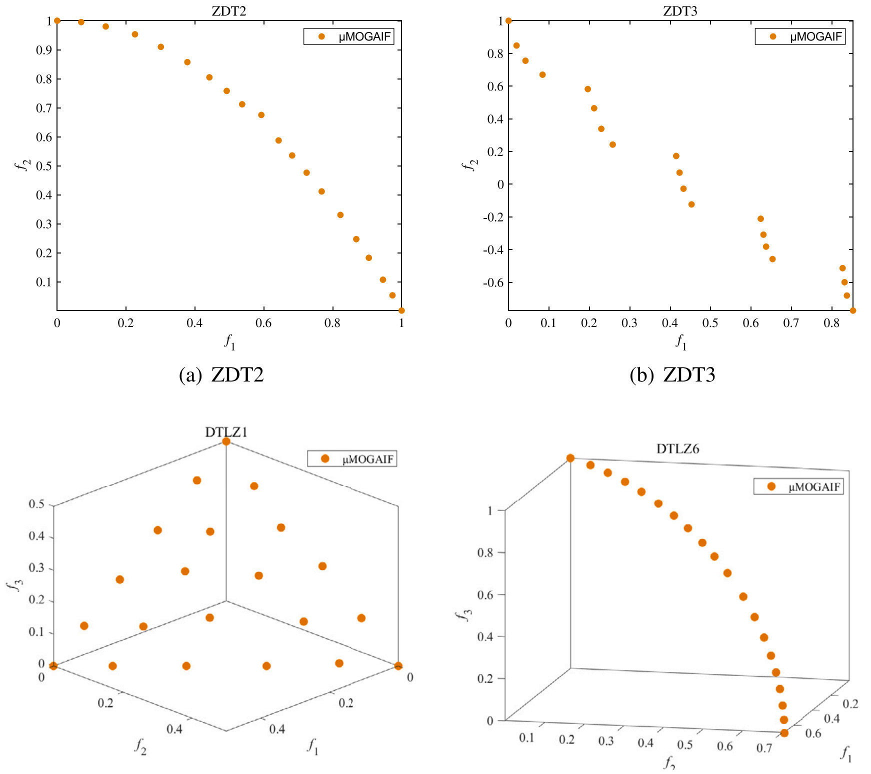
According to the results in Table 6, Two_Arch2 achieved the best values on five test functions, the only one where the μMOGAIF algorithm was inferior on the Friedman test. The test instances of μMOGAIF being weaker than Two_Arch2 are concentrated in the MaF benchmark problems, especially for problems with many decision objectives, such as MaF14 and MaF15. SPEA2+SDE ranked third on the Friedman test and achieved seven best IGD values, eight of which were from MaF benchmark problems.

The statistical results of the Friedman test are performed in Fig. 7, the average rankings on two performance indicators of μMOGAIF, SPEA2+SDE and Two_Arch2 are similar. Compared to bi-objective and tri-objective problems, all algorithms had become weaker in 5-objective problems. Through analyzing the experimental results, it was found that in many-objective problems, the commonly used methods for solving many-objective MOPs are not effective under the premise of using micro populations.

5.5.4. Efficiency analysis of different strategies

To verify the effectiveness of the proposed information fitting strategy, adaptive mating selection, and dual archives strategy, six variants are proposed and compared with the strategies in this paper. The six variants are described below:

- Variant 1: Generate mating pools using only individuals from the essential archive.
- Variant 2: The fitness assignment adopted in the essential archive is based on the original grid ranking.
- Variant 3: The fitness assignment adopted in the essential archive is based on Euclidean distance.
- Variant 4: The operators are selected randomly.
- Variant 5: In updating the compensatory archive, use only the strategy of the $I_{\epsilon+}$.
- Variant 6: In updating the compensatory archive, only crowding-distance assignment is used.



(a) ZDT2 (b) ZDT3

(c) DTLZ1 (d) DTLZ6

Fig. 4. The distribution of the final non-dominated solutions by μMOGAIF on ZDT2, ZDT3, DTLZ1, DTLZ6.

Table 2
Comparisons of average IGD values of $\mu MOGAIF$ with NSGA-III, GrEA, IBEA, Two_Arch2, SPEA2+SDE, MOCPSO on DTLZ, MaF, WFG, ZDT problems run independently 30 times.

Problem	M	D	NSGA-III	GrEA	IBEA	Two_Arch2	SPEA2+SDE	MOCPSO	μMOGAIF
DTLZ1	3	7	1.7832e−1 (1.75e−1) =	3.4939e−1 (1.21e−1) −	3.7316e−1 (6.15e−2) −	5.9603e−1 (6.65e−1) −	8.1538e−2 (8.56e−2) +	2.4227e+1 (4.00e+0) −	1.9811e−1 (2.10e−1)
DTLZ2	3	12	1.2862e−1 (1.20e−5) +	6.5685e−2 (1.88e−3) +	1.6189e−1 (7.71e−3) −	1.4104e−1 (3.84e−3) −	1.5868e−1 (7.88e−3) −	1.8045e−1 (1.48e−2) −	1.3266e−1 (1.66e−3)
DTLZ3	3	12	1.0689e+1 (5.92e+0) =	6.9809e+0 (4.95e+0) =	4.5286e+0 (3.59e+0) +	1.3789e+1 (8.24e+0) −	4.8163e+0 (2.94e+0) +	1.7438e+2 (2.73e+1) −	9.0061e+0 (5.65e+0)
DTLZ4	3	12	6.0406e−1 (2.69e−1) =	7.5317e−1 (2.04e−1) =	5.5136e−1 (3.45e−1) =	6.4724e−1 (3.97e−1) =	7.0991e−1 (2.75e−1) =	1.6835e−1 (1.07e−2) +	6.9107e−1 (3.01e−1)
DTLZ5	3	12	5.6760e−2 (1.18e−2) −	3.6951e−2 (4.55e−4) −	4.5311e−2 (5.44e−3) −	2.8354e−2 (2.36e−3) −	3.7705e−2 (4.00e−3) −	1.6778e−1 (5.41e−3) −	2.1896e−2 (4.14e−4)
DTLZ6	3	12	8.0099e−2 (1.70e−2) −	3.7146e−2 (5.77e−4) −	5.0163e−2 (5.91e−3) −	2.6798e−2 (1.91e−3) −	3.6297e−2 (4.51e−3) −	2.7936e+0 (8.41e−1) −	2.0944e−2 (1.63e−4)
DTLZ7	3	22	3.3478e−1 (1.40e−1) =	6.1608e−1 (2.43e−1) −	4.4989e−1 (2.93e−1) −	2.9993e−1 (1.83e−1) =	3.6370e−1 (1.74e−1) −	5.2871e+0 (1.51e+0) −	3.3100e−1 (2.16e−1)
MaF1	3	12	1.2744e−1 (5.02e−3) −	2.9352e−1 (9.39e−2) −	1.2047e−1 (1.72e−2) −	1.0438e−1 (4.08e−3) =	1.0529e−1 (7.09e−3) =	2.7996e−1 (2.44e−2) −	1.0271e−1 (3.15e−3)
MaF2	3	12	8.5213e−2 (3.29e−3) −	1.3673e−1 (5.43e−3) −	7.8466e−2 (3.46e−3) −	7.5474e−2 (1.61e−3) −	9.1423e−2 (1.15e−2) −	1.2078e−1 (5.46e−3) −	7.2232e−2 (1.57e−3)
MaF3	3	12	1.0336e+2 (7.27e+1) +	3.7002e+3 (1.22e+4) +	5.5331e+3 (1.28e+4) =	2.4521e+2 (2.28e+2) +	4.7933e+1 (5.08e+1) +	3.6579e+4 (7.60e+3) −	1.4191e+4 (2.84e+4)
MaF4	3	12	3.8484e+1 (2.39e+1) +	1.1090e+1 (1.09e+1) +	2.7163e+1 (2.03e+1) +	3.3709e+1 (2.03e+1) +	1.5693e+1 (1.35e+1) +	6.1029e+2 (1.01e+2) −	5.9553e+1 (3.36e+1)
MaF5	3	12	2.7069e+0 (1.64e+0) =	2.7921e+0 (1.68e+0) =	2.7624e+0 (1.97e+0) =	1.8713e+0 (2.00e+0) =	3.0174e+0 (1.75e+0) =	8.9787e−1 (4.31e−2) +	2.9005e+0 (1.86e+0)
MaF6	3	12	6.9274e−2 (2.08e−2) −	3.6614e−2 (1.51e−3) −	6.9162e−2 (1.18e−2) −	2.6798e−2 (8.56e−4) −	3.8225e−2 (5.75e−3) −	4.5097e−1 (2.20e−1) −	2.0926e−2 (1.74e−4)
MaF7	3	22	3.5212e−1 (1.79e−1) −	6.4787e−1 (3.54e−1) −	3.7863e−1 (2.08e−1) −	3.1804e−1 (2.09e−1) −	2.7839e−1 (1.06e−1) −	6.7403e+0 (5.79e+0) −	2.6043e−1 (1.67e−1)
WFG1	3	12	6.8542e−1 (1.35e−1) −	1.0949e+0 (4.95e−1) −	8.8170e−1 (4.39e−1) −	8.1773e−1 (1.45e−1) −	7.1172e−1 (1.69e−1) −	2.1904e+0 (4.26e−1) −	5.1729e−1 (9.13e−2)
WFG2	3	12	3.8247e−1 (5.39e−2) =	1.3686e+0 (1.70e−1) −	4.9693e−1 (2.11e−1) −	4.9142e−1 (2.10e−1) −	4.6312e−1 (5.81e−2) −	5.4511e−1 (3.55e−2) −	4.5592e−1 (2.74e−1)
WFG3	3	12	2.3536e−1 (3.03e−2) −	2.2109e+0 (3.50e−1) −	9.5406e−2 (9.29e−3) +	2.1021e−1 (4.28e−2) −	1.3436e−1 (4.24e−2) +	7.3907e−1 (6.55e−2) −	1.8664e−1 (3.28e−2)
WFG4	3	12	5.1950e−1 (6.43e−4) −	8.1794e−1 (1.93e−1) −	6.2308e−1 (4.48e−2) −	5.3634e−1 (2.30e−2) −	6.7017e−1 (4.50e−2) −	7.2513e−1 (2.59e−2) −	5.1247e−1 (8.05e−3)
WFG5	3	12	5.1290e−1 (2.27e−3) −	7.5997e−1 (2.78e−2) −	6.3679e−1 (3.27e−2) −	5.4948e−1 (6.47e−2) −	6.5329e−1 (7.79e−2) −	6.4983e−1 (1.25e−2) −	5.0220e−1 (9.52e−3)
WFG6	3	12	5.1731e−1 (3.27e−3) =	7.5109e−1 (2.50e−2) −	6.5504e−1 (1.95e−2) −	5.5949e−1 (2.44e−2) −	6.6259e−1 (4.39e−2) −	6.9914e−1 (2.26e−2) −	5.2095e−1 (9.22e−3)
WFG7	3	12	5.2228e−1 (2.87e−3) =	7.8480e−1 (5.35e−2) −	6.5098e−1 (3.36e−2) −	5.4801e−1 (2.66e−2) −	6.7122e−1 (3.85e−2) −	7.3830e−1 (2.14e−2) −	5.2135e−1 (1.10e−2)
WFG8	3	12	5.2765e−1 (4.33e−3) +	7.4389e−1 (3.64e−2) −	6.2166e−1 (1.96e−2) −	5.8618e−1 (1.59e−2) −	6.5327e−1 (3.16e−2) −	8.3565e−1 (6.04e−2) −	5.4326e−1 (1.05e−2)
WFG9	3	12	5.3004e−1 (2.50e−2) −	8.0672e−1 (5.51e−2) −	6.0925e−1 (3.18e−2) −	5.5355e−1 (3.83e−2) −	6.5294e−1 (3.06e−2) −	6.6774e−1 (1.88e−2) −	5.0848e−1 (3.11e−2)
ZDT1	2	30	1.9218e−2 (4.18e−5) +	5.2641e−1 (8.45e−2) −	2.1164e−2 (2.17e−3) −	3.3150e−2 (3.67e−2) −	3.3780e−2 (1.86e−2) −	1.4314e+0 (2.93e−1) −	1.9441e−2 (3.01e−4)
ZDT2	2	30	2.2085e−2 (1.10e−2) −	1.8268e−1 (2.41e−2) −	3.1358e−2 (1.15e−2) −	1.4469e−1 (8.16e−2) −	2.8953e−2 (6.16e−3) −	2.0833e+0 (3.99e−1) −	1.9808e−2 (1.99e−4)
ZDT3	2	30	4.0087e−2 (1.90e−2) −	4.8733e−1 (1.85e−1) −	5.7854e−2 (2.11e−2) −	5.6238e−2 (3.79e−2) −	3.9963e−2 (1.99e−2) −	1.0961e+0 (2.65e−1) −	3.9910e−2 (2.57e−2)
ZDT4	2	10	1.2712e−1 (8.52e−2) +	7.1669e−1 (1.43e−1) −	6.0899e−1 (1.37e−1) −	4.4955e−1 (1.74e−1) −	6.6329e−2 (4.00e−2) +	1.7314e+2 (6.57e+1) −	2.8588e−1 (1.36e−1)
ZDT6	2	10	1.5632e−2 (8.88e−4) =	8.8427e−2 (7.81e−3) −	2.3328e−2 (2.25e−3) −	1.7181e−2 (5.78e−4) −	1.8435e−2 (1.56e−3) −	3.5801e+0 (2.16e+0) −	1.5648e−2 (1.76e−4)
+/−/≈			6/13/9	3/22/3	3/22/3	2/22/4	6/19/3	2/26/0	

M is the number of objectives, and D is the dimensions of decision variables. Each row lists the IGD value(the value in parentheses is the standard deviation) of the algorithm under the problem, and the best value is highlighted in blue. The symbols "+", "−", and "≈" in the Wilcoxon test indicated that the algorithm performed better, worse, and similarly compared to $\mu MOGAIF$, respectively.

Tables 10 and 11 show the statistical results obtained by μMOGAIF and the six variants in 28 instances with bi-objectives or tri-objectives. In the result shown, μMOGAIF achieved the best IGD values on 11 test suites, with the other variants achieving the best values of 2, 0, 3, 7, 4, and 1, respectively. According to the statistical results of the Friedman test, the average ranking of μMOGAIF is lower than all variants. μMOGAIF achieved lower IGD values than variant 1 on most issues except DTLZ5, MaF5, WFG4, WFG5, and ZDT4. The individuals in the mating pool in variant 1 are only sourced from the essential archive, which represents the impact of the compensatory archive on population evolution. To demonstrate the effectiveness of grid ranking combined with reference points, different methods are used in the allocation of individual fitness in variant 2 and variant 3. As a common convergence fitness value, it can be observed from variant 2 that the original grid ranking is not suitable for the proposed algorithm. However, after adjusting the fitness allocation of grid ranking, the performance of the algorithm has significantly improved. Information fitting strategy plays a vital role in operator selection, variant 4 is set to validate the operator selection, while variant 5 and variant 6 are set for the strategy selection in the compensatory archive. In variant 5 and variant 6, the compensatory archive is only updated by one strategy throughout the entire evolutionary process. It can be seen that the

Table 3
Comparisons of average spread values of $\mu MOGAIF$ with NSGA-III, GrEA, IBEA, Two_Arch2, SPEA2+SDE, MOCPSO on DTLZ, MaF, WFG, ZDT problems run 30 times.

Problem	M	D	NSGA-III	GrEA	IBEA	Two_Arch2	SPEA2+SDE	MOCPSO	μMOGAIF
DTLZ1	3	7	2.5125e−1 (2.55e−1) =	1.0380e+0 (2.20e−1) −	1.6143e+1 (5.08e+1) −	4.5713e−1 (1.12e−1) −	3.4336e−1 (6.01e−2) −	4.6837e−1 (1.04e−1) −	2.5842e−1 (1.95e−1)
DTLZ2	3	12	1.9283e−1 (4.46e−4) −	4.6384e−1 (4.04e−2) −	5.5240e−1 (8.48e−2) −	3.4103e−1 (6.35e−2) −	4.2423e−1 (5.93e−2) −	2.1878e−1 (1.78e−2) −	8.4972e−2 (2.29e−2)
DTLZ3	3	12	9.8627e−1 (3.82e−1) =	1.1058e+0 (2.62e−1) =	1.4717e+0 (2.38e−1) −	8.8389e−1 (2.29e−1) =	8.2467e−1 (1.48e−1) =	6.0431e−1 (5.88e−2) +	1.0346e+0 (4.94e−1)
DTLZ4	3	12	8.8897e−1 (3.23e−1) −	1.0017e+0 (2.36e−2) −	7.9665e−1 (2.32e−1) =	7.2267e−1 (3.47e−1) =	8.6113e−1 (1.93e−1) =	2.0158e−1 (2.70e−2) +	7.1727e−1 (3.41e−1)
DTLZ5	3	12	1.1352e+0 (1.30e−1) −	1.0863e+0 (4.38e−2) −	7.9814e−1 (7.44e−2) −	5.9411e−1 (3.93e−2) −	7.1536e−1 (9.09e−2) −	1.0947e+0 (1.56e−1) −	1.3560e−1 (3.21e−2)
DTLZ6	3	12	1.6116e+0 (2.15e−1) −	1.0779e+0 (3.76e−2) −	9.0942e−1 (1.71e−1) −	4.3761e−1 (6.10e−2) −	7.2897e−1 (8.36e−2) −	6.5356e−1 (1.28e−1) −	1.0362e−1 (2.99e−2)
DTLZ7	3	22	9.7908e−1 (1.29e−1) −	9.6768e−1 (2.05e−1) −	6.1665e−1 (1.37e−1) −	4.3081e−1 (1.03e−1) −	5.4687e−1 (1.07e−1) −	1.0109e+0 (6.00e−2) −	3.0560e−1 (1.32e−1)
MaF1	3	12	9.2072e−1 (1.74e−1) −	7.4647e−1 (9.12e−2) −	4.7535e−1 (5.91e−2) −	2.6429e−1 (7.06e−2) −	2.8557e−1 (6.84e−2) −	1.4810e−1 (1.13e−1) =	1.3902e−1 (5.34e−2)
MaF2	3	12	1.0382e+0 (1.41e−1) −	3.1496e−1 (2.68e−2) −	3.1169e−1 (8.43e−2) −	2.9941e−1 (8.42e−2) −	2.9016e−1 (7.43e−2) −	2.7345e−1 (3.68e−2) −	1.1421e−1 (1.90e−2)
MaF3	3	12	1.5990e+0 (6.69e−1) −	1.0874e+0 (2.53e−1) =	1.0000e+0 (1.05e−4) =	1.1040e+0 (4.84e−1) =	8.5474e−1 (2.55e−1) =	1.6149e+0 (1.87e+0) =	1.1215e+0 (5.33e−1)
MaF4	3	12	1.0876e+0 (2.94e−1) −	9.0109e−1 (1.17e−1) −	1.0353e+0 (8.70e−2) −	5.2545e−1 (8.62e−2) +	6.7795e−1 (1.08e−1) −	8.1706e−1 (2.00e−1) −	5.5151e−1 (3.27e−1)
MaF5	3	12	9.3693e−1 (2.86e−1) −	9.5337e−1 (1.15e−1) −	7.5618e−1 (2.69e−1) =	5.0074e−1 (3.13e−1) =	8.3964e−1 (2.21e−1) =	4.4681e−1 (6.42e−2) =	6.2413e−1 (3.64e−1)
MaF6	3	12	1.2158e+0 (1.89e−1) −	1.0548e+0 (1.02e−1) −	1.1990e+0 (2.26e−1) −	4.7438e−1 (4.82e−2) −	7.4094e−1 (9.11e−2) −	9.4607e−1 (1.01e+0) −	9.0172e−2 (2.67e−2)
MaF7	3	22	9.4022e−1 (1.63e−1) −	8.7115e−1 (2.11e−1) −	5.8354e−1 (9.11e−2) −	4.1138e−1 (1.26e−1) −	5.5452e−1 (9.86e−2) −	9.9446e−1 (5.04e−2) −	2.4110e−1 (1.05e−1)
WFG1	3	12	5.9946e−1 (8.40e−2) −	9.6085e−1 (1.33e−1) −	8.1895e−1 (8.41e−2) −	5.7686e−1 (4.41e−2) −	7.2643e−1 (7.86e−2) −	1.8082e+0 (1.38e+0) −	3.5506e−1 (5.90e−2)
WFG2	3	12	4.0266e−1 (4.79e−2) −	1.0445e+0 (6.73e−2) −	7.5004e−1 (6.88e−2) −	4.4232e−1 (1.27e−1) −	7.5208e−1 (8.47e−2) −	6.0986e−1 (5.82e−2) −	2.2919e−1 (1.09e−1)
WFG3	3	12	9.3956e−1 (1.85e−1) −	1.0207e+0 (1.99e−2) −	2.9417e−1 (8.93e−2) −	2.8723e−1 (7.03e−2) −	3.8736e−1 (9.19e−2) −	5.9972e−1 (8.75e−2) −	1.4359e−1 (4.29e−2)
WFG4	3	12	3.3452e−1 (5.19e−3) −	6.0493e−1 (9.10e−2) −	5.0199e−1 (1.02e−1) −	3.1179e−1 (9.49e−2) −	4.4971e−1 (6.90e−2) −	2.8386e−1 (8.01e−2) −	1.0310e−1 (2.17e−2)
WFG5	3	12	3.4579e−1 (2.04e−2) −	5.9768e−1 (7.38e−2) −	4.9632e−1 (8.12e−2) −	4.2487e−1 (9.96e−2) −	4.3946e−1 (9.68e−2) −	4.1879e−1 (1.45e−2) −	1.1290e−1 (2.39e−2)
WFG6	3	12	3.4697e−1 (5.52e−3) −	6.2611e−1 (5.72e−2) −	5.1392e−1 (9.18e−2) −	3.2000e−1 (7.46e−2) −	4.8150e−1 (8.45e−2) −	3.8910e−1 (5.70e−2) −	1.1210e−1 (2.61e−2)
WFG7	3	12	3.3641e−1 (5.16e−3) −	6.0928e−1 (6.19e−2) −	4.9948e−1 (8.32e−2) −	3.1305e−1 (7.11e−2) −	4.1913e−1 (7.07e−2) −	3.6389e−1 (6.39e−2) −	1.0109e−1 (2.07e−2)
WFG8	3	12	3.3164e−1 (3.15e−2) −	6.5599e−1 (1.20e−1) −	4.2877e−1 (6.39e−2) −	3.1678e−1 (9.91e−2) −	4.3995e−1 (8.32e−2) −	4.5294e−1 (5.34e−2) −	1.2375e−1 (3.19e−2)
WFG9	3	12	4.0554e−1 (3.59e−2) −	5.8831e−1 (7.87e−2) −	4.9537e−1 (6.41e−2) −	3.9036e−1 (1.03e−1) −	4.3455e−1 (6.64e−2) −	4.4866e−1 (3.34e−2) −	1.4281e−1 (3.31e−2)
ZDT1	2	30	3.1290e−1 (1.18e−3) −	9.9590e−1 (3.33e−2) −	4.0477e−1 (4.71e−2) −	3.7784e−1 (7.65e−2) −	5.1946e−1 (8.72e−2) −	8.0013e−1 (4.26e−2) −	1.0681e−1 (2.80e−2)
ZDT2	2	30	2.1795e−1 (1.27e−1) −	8.0569e−1 (1.02e−1) −	4.6800e−1 (7.99e−2) −	6.5743e−1 (1.56e−1) −	4.4591e−1 (9.12e−2) −	9.2142e−1 (4.48e−2) −	8.8166e−2 (1.98e−2)
ZDT3	2	30	4.6581e−1 (1.13e−1) −	1.0075e+0 (9.76e−2) −	9.0005e−1 (6.96e−2) −	4.0566e−1 (1.34e−1) −	6.5775e−1 (9.66e−2) −	7.9486e−1 (6.22e−2) −	2.3251e−1 (1.10e−1)
ZDT4	2	10	7.0017e−1 (1.62e−1) =	1.3035e+0 (4.18e−1) −	1.1162e+0 (9.15e−2) −	7.9098e−1 (1.07e−1) −	5.6597e−1 (1.20e−1) +	9.9858e−1 (5.03e−3) −	6.7393e−1 (1.25e−1)
ZDT6	2	10	1.2709e−1 (9.01e−3) −	3.3526e−1 (1.41e−1) −	5.2691e−1 (1.33e−1) −	3.2738e−1 (2.40e−2) −	2.9505e−1 (7.31e−2) −	1.0226e+0 (4.34e−1) −	9.9473e−2 (3.43e−2)
+/−/≈			0/25/3	0/26/2	0/25/3	1/23/4	1/23/4	2/23/3	

M is the number of objectives, and D is the dimensions of decision variables. Each row lists the Spread value (the value in parentheses is the standard deviation) of the algorithm under the problem, and the best value is highlighted in blue. The symbols "+", "−", and "≈" in the Wilcoxon test indicated that the algorithm performed better, worse, and similarly compared to $\mu MOGAIF$, respectively.

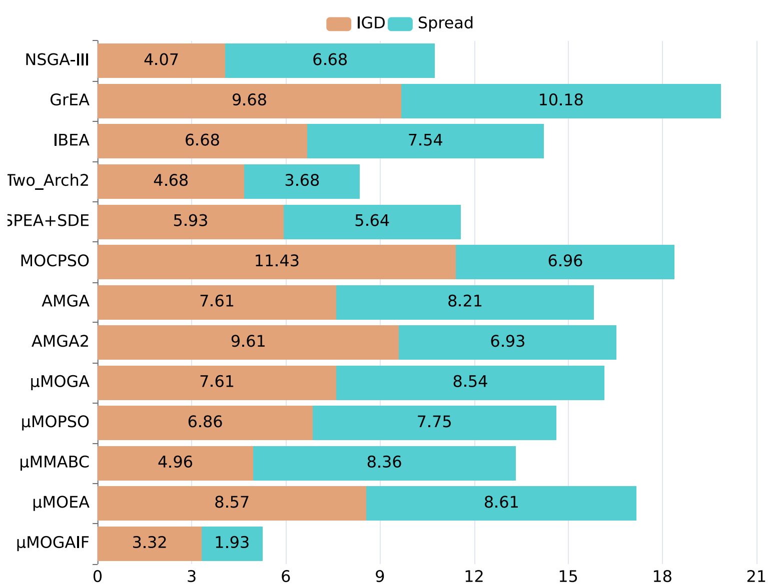


Fig. 5. Ranking histogram of μMOGAIF and other 12 algorithms on problems DTLZ, MaF, WFG, ZDT. Average ranks were obtained from the Friedman test using the IGD and Spread metric.

comprehensive application of update strategies is more effective. For the three crossover operators mentioned in this article, the operator selection in variant 4 is based on random selection. In addition, to further explore the effectiveness of operator selection, we examined the impact of each crossover operator on the performance of μMOGAIF by excluding it from the set of operators used by the algorithm. The variants are described below:

- μMOGAIF-1: Evolutionary processes do not involve SBX operator
- μMOGAIF-2: Evolutionary processes do not involve DE operator
- μMOGAIF-3: Evolutionary processes do not involve GA operator

The comprehensive performance of μMOGAIF and the other three variants on 21 different problems is presented in Table 12. It is evident that μMOGAIF using all crossover operators exhibits superior capabilities in solving various problems, and the impact of the three crossover operators is not significantly different. The three crossover operators selected in the article are all very effective. However, removing any operator may diminish the performance of the algorithm. μMOGAIF achieved optimal values on 13 problems, and other variants achieved on 3, 3, and 2 problems, respectively. It is difficult to find a specific operator that is most suitable for the whole population evolution process. For instance, the variants without DE operators performed better than the other variants on DTLZ4, while μMOGAIF that integrates the DE operator greatly improves its performance.

5.6. Simulations based on low-power microprocessor

In this subsection, to verify the feasibility of the proposed μMOGAIF running on the low-power microprocessor, the developed microcontroller ESP32-WROOM is used as the experimental platform. The ESP32-WROOM is a new generation of WiFi and Bluetooth dual-mode, dual-core wireless communication chips recently released by Espressif. The chip embedded in ESP32-WROOM utilizes a low-power microprocessor that can replace the CPU to conserve power when executing tasks that do not demand much computing power. The ESP32-WROOM is applied in multiple real-world scenarios because of its low power

Table 4
Comparisons of average IGD values of AMGA, AMGA2, μMMOGA, μMMOPSO, μMMABC, μMOEA, μMOGAIF on DTLZ, MaF, WFG, ZDT test suites run independently 30 times.

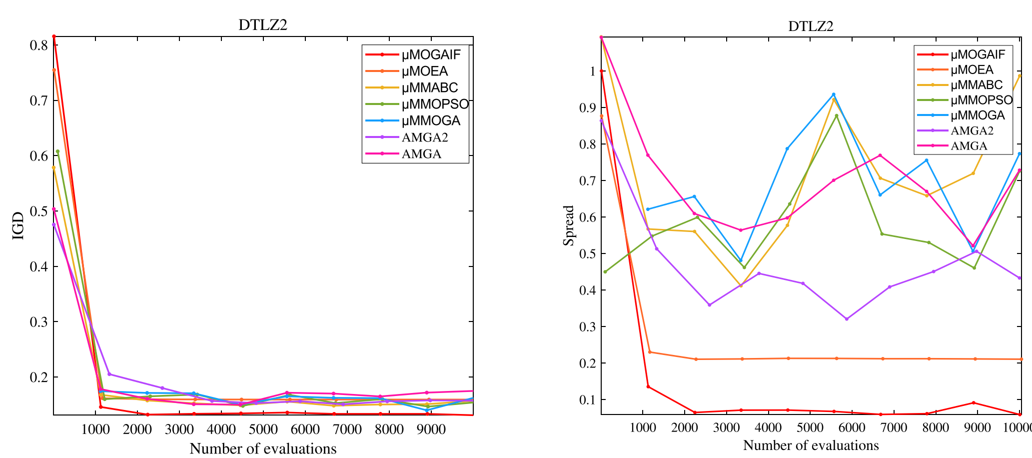
Problem	M	D	AMGA	AMGA2	μMMOGA	μMMOPSO	μMMABC	μMOEA	μMOGAIF
DTLZ1	3	7	6.5826e−1 (5.57e−1) −	3.8572e+0 (2.51e+0) −	4.4267e−1 (3.50e−1) −	1.7281e+0 (1.84e+0) −	6.4270e−2 (8.51e−3) =	7.6132e−1 (1.07e+0) −	1.8857e−1 (2.02e−1)
DTLZ2	3	12	1.6024e−1 (9.26e−3) −	1.6582e−1 (1.93e−2) −	1.5895e−1 (8.44e−3) −	1.5800e−1 (9.59e−3) −	1.4803e−1 (5.64e−3) −	1.5916e−1 (9.26e−5) −	1.3267e−1 (1.60e−3)
DTLZ3	3	12	2.1594e+1 (1.14e+1) −	6.7989e+1 (3.89e+1) −	1.2326e+1 (6.02e+0) −	5.5174e+1 (2.95e+1) −	1.1219e+1 (1.60e+1) =	2.0931e+1 (1.46e+1) −	8.8229e+0 (5.57e+0)
DTLZ4	3	12	8.9271e−1 (2.03e−1) −	6.7272e−1 (3.60e−1) +	7.3451e−1 (3.57e−1) =	2.1114e−1 (2.00e−1) +	1.5142e−1 (8.04e−3) +	8.4424e−1 (2.64e−1) −	6.8966e−1 (2.91e−1)
DTLZ5	3	12	2.7659e−2 (1.73e−3) −	2.9495e−2 (4.01e−3) −	2.9104e−2 (2.15e−3) −	3.0839e−2 (3.31e−3) −	6.9303e−2 (1.21e−2) −	2.9798e−2 (7.59e−4) −	2.1863e−2 (4.20e−4)
DTLZ6	3	12	2.6753e−2 (1.69e−3) −	2.5532e−2 (1.33e−3) −	2.6133e−2 (4.83e−1) −	3.2409e−2 (3.17e−3) −	8.7273e−2 (5.86e−3) −	3.4307e−2 (2.38e−3) −	2.0940e−2 (1.59e−4)
DTLZ7	3	22	7.8299e−1 (1.04e−1) −	5.8836e−1 (2.93e−1) −	4.0255e−1 (2.27e−1) −	3.6029e−1 (2.37e−1) =	2.6525e−1 (3.46e−2) =	3.0354e−1 (3.71e−2) =	3.2904e−1 (2.08e−1)
MaF1	3	12	1.3443e−1 (1.01e−2) −	1.6102e−1 (1.85e−2) −	1.3427e−1 (9.08e−3) −	1.3853e−1 (1.11e−2) −	1.7492e−1 (1.05e−2) −	1.2912e−1 (2.93e−3) −	1.0285e−1 (3.08e−3)
MaF2	3	12	1.1909e−1 (1.13e−2) −	1.0897e−1 (6.09e−3) −	1.0900e−1 (1.34e−2) −	1.0339e−1 (9.44e−3) −	1.0892e−1 (1.03e−2) −	1.0040e−1 (9.80e−3) −	7.2188e−2 (1.52e−3)
MaF3	3	12	8.0787e+2 (7.89e+2) +	8.7371e+3 (5.13e+3) +	4.5358e+2 (7.10e+2) +	5.0556e+3 (4.30e+3) =	1.0864e+3 (4.65e+3) +	3.3964e+2 (3.81e+2) +	1.3526e+4 (2.70e+4)
MaF4	3	12	8.9428e+1 (4.96e+1) −	2.5216e+2 (1.05e+2) −	4.8411e+1 (2.87e+1) =	1.6355e+2 (9.49e+1) −	3.1789e+1 (5.39e+1) +	6.7154e+1 (3.99e+1) −	5.8364e+1 (3.33e+1)
MaF5	3	12	4.4664e+0 (1.29e+0) −	3.9307e+0 (1.65e+0) −	4.1907e+0 (1.59e+0) =	8.5887e−1 (7.65e−1) +	7.5426e−1 (2.52e−2) +	4.0720e+0 (1.66e+0) −	2.9868e+0 (1.85e+0)
MaF6	3	12	2.8831e−2 (2.21e−3) −	1.3633e−1 (2.35e−1) −	2.9254e−2 (2.78e−3) −	2.7177e−2 (2.71e−3) −	7.2389e−2 (1.16e−2) −	3.3584e−2 (1.97e−2) −	2.0922e−2 (1.86e−4)
MaF7	3	22	7.5515e−1 (1.43e−1) −	5.7835e−1 (2.90e−1) −	4.3228e−1 (2.66e−1) −	3.1467e−1 (1.83e−1) =	2.6572e−1 (4.19e−2) =	3.0805e−1 (3.08e−2) =	2.7292e−1 (1.63e−1)
WFG1	3	12	1.0802e+0 (1.67e−1) −	1.6281e+0 (5.09e−2) −	9.9187e−1 (1.33e−1) −	8.8301e−1 (7.45e−2) −	2.0582e+0 (9.02e−2) −	1.3519e+0 (1.14e−1) −	5.1859e−1 (8.78e−2)
WFG2	3	12	6.0536e−1 (7.77e−2) −	6.3480e−1 (6.96e−2) −	6.1215e−1 (1.72e−1) −	5.6008e−1 (6.89e−2) −	4.4466e−1 (2.05e−2) +	6.0513e−1 (1.24e−1) −	4.5208e−1 (2.62e−1)
WFG3	3	12	2.8450e−1 (3.98e−2) −	5.3399e−1 (9.86e−2) −	3.3695e−1 (4.98e−2) −	3.4346e−1 (6.51e−2) −	5.9274e−1 (6.26e−2) −	6.9519e−1 (1.94e−1) −	1.8464e−1 (3.17e−2)
WFG4	3	12	5.7772e−1 (2.54e−2) −	7.2404e−1 (3.88e−2) −	5.9606e−1 (3.69e−2) −	5.9790e−1 (2.51e−2) −	5.7960e−1 (1.76e−2) −	1.0326e+0 (9.05e−2) −	5.1309e−1 (8.00e−3)
WFG5	3	12	6.0026e−1 (3.16e−2) −	7.2040e−1 (2.76e−1) −	6.0476e−1 (3.85e−2) −	6.2661e−1 (3.62e−2) −	5.6784e−1 (2.09e−2) −	8.5682e−1 (2.11e−1) −	5.0274e−1 (9.19e−3)
WFG6	3	12	6.1467e−1 (3.54e−2) −	8.1213e−1 (5.86e−2) −	6.2721e−1 (3.11e−2) −	6.3450e−1 (5.06e−2) −	6.0504e−1 (1.37e−2) −	6.7400e−1 (3.95e−2) −	5.2142e−1 (8.97e−3)
WFG7	3	12	6.3124e−1 (5.18e−2) −	8.5515e−1 (6.12e−2) −	6.6838e−1 (7.07e−2) −	5.9656e−1 (3.57e−2) −	6.0161e−1 (1.54e−2) −	7.3529e−1 (8.15e−2) −	5.2127e−1 (1.06e−2)
WFG8	3	12	6.8202e−1 (3.43e−2) −	7.8021e−1 (3.17e−2) −	6.8438e−1 (2.84e−2) −	6.6688e−1 (3.19e−2) −	6.0245e−1 (8.97e−3) −	1.1573e+0 (2.84e−2) −	5.4290e−1 (1.01e−2)
WFG9	3	12	6.2096e−1 (7.10e−2) −	6.9696e−1 (4.27e−2) −	6.5616e−1 (6.25e−2) −	6.3139e−1 (4.27e−2) −	5.7462e−1 (1.93e−2) −	1.0568e+0 (1.25e−1) −	5.0722e−1 (2.98e−2)
ZDT1	2	30	2.3757e−2 (1.96e−3) −	2.3305e−2 (1.24e−3) −	2.4708e−2 (1.58e−3) −	2.8322e−2 (2.77e−3) −	2.0193e−2 (1.19e−5) −	4.8871e−2 (4.73e−3) −	1.9406e−2 (3.07e−4)
ZDT2	2	30	2.4075e−2 (1.41e−3) −	2.4747e−2 (6.59e−3) −	2.5813e−2 (1.76e−3) −	3.0709e−2 (3.57e−3) −	1.9838e−2 (6.64e−6) =	4.1742e−2 (2.95e−2) −	1.9827e−2 (2.02e−4)
ZDT3	2	30	9.4008e−2 (6.86e−2) −	7.1878e−2 (7.13e−2) −	5.7632e−2 (3.53e−2) −	3.5723e−2 (6.11e−3) =	3.3692e−2 (1.34e−3) =	7.1189e−2 (1.55e−2) −	4.1597e−2 (2.88e−2)
ZDT4	2	10	3.2674e−2 (2.14e−2) +	2.6128e−2 (1.50e−2) +	5.8778e−2 (4.71e−2) +	2.3941e−1 (1.53e−1) =	2.0221e−2 (6.62e−5) +	2.7368e−2 (8.83e−3) +	2.7749e−1 (1.32e−1)
ZDT6	2	10	2.0197e−2 (2.27e−3) −	3.2677e−1 (3.70e−1) −	2.0645e−2 (1.27e−3) −	2.5040e−2 (2.52e−3) −	1.5638e−2 (3.39e−6) +	1.6172e−2 (3.03e−6) −	1.5640e−2 (1.68e−4)
+/−/≈			2/26/0	3/25/0	2/23/3	2/21/5	7/15/6	2/23/3	

M is the number of objectives, and D is the dimensions of decision variables. Each row lists the IGD value(the value in parentheses is the standard deviation) of the algorithm under the problem, and the best value is highlighted in blue. The symbols "+", "−", and "≈" in the Wilcoxon test indicated that the algorithm performed better, worse, and similarly compared to $\mu MOGAIF$, respectively.

Table 5
Comparisons of average spread values of AMGA, AMGA2, μMMOGA, μMMOPSO, μMMABC, μMOEA, μMOGAIF on DTLZ, MaF, WFG, ZDT test suites run 30 times.

Problem	M	D	AMGA	AMGA2	μMMOGA	μMMOPSO	μMMABC	μMOEA	μMOGAIF
DTLZ1	3	7	7.1185e−1 (1.98e−1) −	8.5713e−1 (1.84e−1) −	7.4297e−1 (1.30e−1) −	6.8750e−1 (1.03e−1) −	5.8789e−1 (3.84e−1) −	3.4574e−1 (2.36e−1) −	2.5556e−1 (1.96e−1)
DTLZ2	3	12	7.0949e−1 (1.42e−1) −	4.9192e−1 (1.95e−1) −	6.7144e−1 (1.32e−1) −	6.1167e−1 (1.04e−1) −	5.6475e−1 (2.25e−1) −	2.1098e−1 (4.64e−3) −	8.2467e−2 (2.43e−2)
DTLZ3	3	12	1.0534e+0 (2.13e−1) =	1.0467e+0 (2.61e−1) =	9.7566e−1 (1.56e−1) =	8.5876e−1 (1.20e−1) =	1.3868e+0 (5.10e−1) −	6.7153e−1 (1.45e−1) +	9.9473e−1 (4.86e−1)
DTLZ4	3	12	8.6009e−1 (2.19e−1) =	8.3547e−1 (2.97e−1) −	9.0450e−1 (1.75e−1) −	6.5813e−1 (1.58e−1) =	6.2198e−1 (2.23e−1) +	9.2840e−1 (2.10e−1) −	7.1633e−1 (3.30e−1)
DTLZ5	3	12	6.8134e−1 (1.29e−2) −	5.0837e−1 (1.18e−1) −	7.1362e−1 (1.40e−1) −	8.1816e−1 (1.86e−1) −	1.3513e+0 (1.96e−1) −	2.3146e−1 (5.57e−2) −	1.3170e−1 (3.32e−2)
DTLZ6	3	12	6.6809e−1 (1.31e−1) −	5.1674e−1 (5.13e−2) −	1.0034e+0 (1.85e−1) −	9.9589e−1 (1.83e−1) −	1.9826e+0 (2.29e−1) −	6.1620e−1 (1.27e−1) −	1.0111e−1 (3.07e−2)
DTLZ7	3	22	8.1186e−1 (8.05e−2) −	6.3237e−1 (1.67e−1) −	7.1451e−1 (1.45e−1) −	7.3927e−1 (1.22e−1) −	9.9379e−1 (1.58e−1) −	1.6468e+0 (4.51e−1) −	3.0107e−1 (1.26e−1)
MaF1	3	12	4.5932e−1 (1.23e−1) −	3.7624e−1 (8.57e−2) −	4.5791e−1 (1.19e−1) −	4.4945e−1 (1.22e−1) −	1.6676e+0 (1.90e−1) −	6.7719e−1 (4.33e−2) −	1.4327e−1 (5.85e−2)
MaF2	3	12	5.9695e−1 (9.83e−2) −	4.4587e−1 (8.87e−2) −	6.3919e−1 (1.20e−1) −	6.5766e−1 (1.14e−1) −	8.1246e−1 (1.82e−1) −	6.0523e−1 (3.68e−1) −	1.1416e−1 (1.81e−2)
MaF3	3	12	1.2048e+0 (4.01e−1) =	1.2744e+0 (4.09e−1) =	1.0504e+0 (4.27e−1) =	7.9047e−1 (2.01e−1) =	1.7535e+0 (5.72e−1) −	1.6605e+0 (1.26e−1) −	1.1176e+0 (5.60e−1)
MaF4	3	12	7.4769e−1 (9.97e−2) −	6.7376e−1 (1.08e−1) −	6.9886e−1 (1.11e−1) −	6.7495e−1 (6.37e−2) −	1.6931e+0 (4.29e−1) −	1.2775e+0 (1.74e−2) −	5.4938e−1 (3.11e−1)
MaF5	3	12	7.9016e−1 (2.31e−1) −	9.2078e−1 (2.27e−1) −	9.3569e−1 (1.54e−1) −	6.4981e−1 (1.50e−1) =	8.3956e−1 (1.48e−1) =	8.8956e−1 (2.30e−1) −	6.4146e−1 (3.61e−1)
MaF6	3	12	7.5788e−1 (1.40e−1) −	7.4092e−1 (6.56e−1) −	7.3449e−1 (2.16e−1) −	5.7482e−1 (1.52e−1) −	1.3969e+0 (1.54e−1) −	3.2120e−1 (3.95e−1) −	9.1328e−2 (2.91e−2)
MaF7	3	22	8.1292e−1 (8.31e−2) −	6.3395e−1 (1.44e−1) −	7.1176e−1 (1.42e−1) −	7.2662e−1 (1.51e−1) −	1.0036e+0 (1.92e−1) −	1.6573e+0 (4.07e−1) −	2.5141e−1 (1.04e−1)
WFG1	3	12	7.2058e−1 (1.15e−1) −	8.3741e−1 (1.45e−1) −	7.0933e−1 (8.85e−2) −	6.6189e−1 (9.42e−2) −	1.3432e+0 (2.19e−1) −	1.0762e+0 (2.04e−1) −	3.5503e−1 (5.81e−2)
WFG2	3	12	6.4329e−1 (1.37e−1) −	5.7338e−1 (1.10e−1) −	6.4824e−1 (1.17e−1) −	6.0483e−1 (1.08e−1) −	5.6301e−1 (8.84e−2) −	9.2493e−1 (4.40e−1) −	2.2716e−1 (1.07e−1)
WFG3	3	12	5.8804e−1 (8.47e−2) −	5.1868e−1 (3.21e−1) −	7.1216e−1 (1.53e−1) −	6.2043e−1 (1.74e−1) −	9.9963e−1 (1.44e−1) −	1.8192e+0 (6.10e−1) −	1.4427e−1 (4.19e−2)
WFG4	3	12	5.3994e−1 (1.34e−1) −	5.4146e−1 (1.09e−1) −	5.7280e−1 (1.25e−1) −	5.3384e−1 (1.07e−1) −	4.7111e−1 (1.53e−1) −	1.2419e+0 (3.43e−1) −	1.0109e−1 (2.42e−2)
WFG5	3	12	7.1100e−1 (8.98e−2) −	7.1573e−1 (4.01e−1) −	6.4899e−1 (1.16e−1) −	7.6773e−1 (1.32e−1) −	6.2625e−1 (1.84e−1) −	8.2104e−1 (4.27e−1) −	1.1420e−1 (2.70e−2)
WFG6	3	12	6.4927e−1 (1.39e−1) −	5.1674e−1 (1.12e−1) −	6.5887e−1 (1.62e−1) −	6.4295e−1 (1.75e−1) −	4.1606e−1 (7.75e−2) −	4.3430e−1 (1.96e−1) −	1.1372e−1 (2.63e−2)
WFG7	3	12	6.6814e−1 (1.44e−1) −	5.0228e−1 (7.95e−2) −	6.8228e−1 (1.55e−1) −	6.0144e−1 (1.48e−1) −	4.5163e−1 (1.11e−1) −	6.1412e−1 (3.44e−1) −	1.0257e−1 (2.08e−2)
WFG8	3	12	7.9285e−1 (1.18e−1) −	5.1632e−1 (7.64e−2) −	7.7526e−1 (1.19e−1) −	6.2079e−1 (1.15e−1) −	4.3182e−1 (7.77e−2) −	1.7331e+0 (2.34e−2) −	1.2176e−1 (3.13e−2)
WFG9	3	12	5.9241e−1 (1.05e−1) −	5.2876e−1 (6.77e−2) −	6.2749e−1 (1.12e−1) −	6.3341e−1 (1.12e−1) −	5.6348e−1 (1.57e−1) −	1.5730e+0 (2.71e−1) −	1.4007e−1 (3.26e−2)
ZDT1	2	30	4.8793e−1 (1.67e−1) −	4.6428e−1 (8.87e−2) −	4.7681e−1 (1.24e−1) −	7.7485e−1 (1.28e−1) −	3.1318e−1 (2.88e−3) −	4.0334e−1 (2.59e−2) −	1.0946e−1 (2.91e−2)
ZDT2	2	30	4.4062e−1 (9.62e−2) −	4.7756e−1 (1.28e−1) −	5.4639e−1 (1.83e−1) −	8.2867e−1 (1.47e−1) −	1.6455e−1 (2.18e−3) −	3.7425e−1 (1.69e−1) −	8.9688e−2 (2.02e−2)
ZDT3	2	30	5.6100e−1 (1.45e−1) −	5.4908e−1 (1.34e−1) −	5.6876e−1 (1.48e−1) −	6.5591e−1 (1.75e−1) −	6.6849e−1 (1.13e−1) −	1.0254e+0 (1.15e−1) −	2.3388e−1 (1.19e−1)
ZDT4	2	10	4.8339e−1 (1.64e−1) +	3.0713e−1 (1.32e−1) +	6.4844e−1 (1.69e−1) =	6.7156e−1 (1.58e−1) =	3.1497e−1 (4.59e−3) +	3.3455e−1 (1.87e−2) +	6.6304e−1 (1.23e−1)
ZDT6	2	10	6.9296e−1 (2.48e−1) −	7.2036e−1 (2.20e−1) −	6.6962e−1 (1.51e−1) −	1.0028e+0 (1.42e−1) −	1.2055e−1 (1.99e−3) −	1.5482e−1 (2.03e−3) −	9.8727e−2 (3.26e−2)
+/−/≈			1/24/3	1/25/2	0/25/3	0/23/5	2/25/1	2/26/0	

M is the number of objectives, and D is the dimensions of decision variables. Each row lists the Spread value (the value in parentheses is the standard deviation) of the algorithm under the problem, and the best value is highlighted in blue. The symbols "+", "−", and "≈" in the Wilcoxon test indicated that the algorithm performed better, worse, and similarly compared to $\mu MOGAIF$, respectively.



(a) IGD (b) Spread

Fig. 6. The performance comparison curve of the comparison algorithms based on IGD and Spread indicators.

Fig. 7. Ranking histogram of μMOGAIF and other 12 algorithms on 5-objective problems. Average ranks were obtained from the Friedman test using the IGD and Spread metric.

Table 6
Comparisons of average IGD values of $\mu MOGAIF$ with NSGA-III, GrEA, IBEA, Two_Arch2, SPEA2+SDE, MOCPSO on 5-objective problems run independently for 30 times.

Problem	M	D	NSGA-III	GrEA	IBEA	Two_Arch2	SPEA2+SDE	MOCPSO	μMOGAIF
MaF1	5	14	2.7789e−1 (1.72e−2) −	3.1369e−1 (7.89e−2) −	2.6930e−1 (3.32e−2) =	2.3014e−1 (8.13e−3) +	2.5489e−1 (1.46e−2) =	4.8957e−1 (7.16e−2) −	2.5844e−1 (1.03e−2)
MaF2	5	14	2.1335e−1 (6.62e−3) −	4.0377e−1 (4.33e−2) −	2.4247e−1 (1.38e−2) −	2.0433e−1 (9.40e−3) −	2.4624e−1 (1.06e−2) −	2.2469e−1 (7.44e−3) −	1.9671e−1 (4.15e−3)
MaF3	5	14	4.1884e+2 (5.28e+2) +	1.8035e+4 (3.21e+4) +	1.5752e+4 (3.10e+4) +	1.9920e+4 (6.00e+4) +	6.5603e+1 (8.52e+1) +	3.4282e+4 (7.21e+3) +	1.2726e+6 (2.94e+6)
MaF4	5	14	1.6966e+2 (1.32e+2) +	7.6659e+1 (6.26e+1) +	1.2242e+2 (1.31e+2) +	2.1716e+2 (9.75e+1) +	5.4817e+1 (4.62e+1) +	2.8329e+3 (4.29e+2) −	3.3015e+2 (2.03e+2)
MaF5	5	14	4.9208e+0 (1.00e+0) +	9.7636e+0 (6.76e+0) −	7.2230e+0 (3.31e+0) −	6.1514e+0 (3.63e+0) =	1.0266e+1 (7.07e+0) −	6.8242e+0 (5.13e−1) −	5.0283e+0 (1.63e+0)
MaF6	5	14	1.1352e−1 (3.72e−2) −	5.5810e−2 (3.67e−3) −	6.2204e−2 (1.03e−2) −	3.7370e−2 (6.98e−3) −	3.6532e−2 (5.87e−3) −	1.2480e+0 (1.09e+0) −	2.0950e−2 (2.14e−4)
MaF7	5	24	6.8369e−1 (1.50e−1) −	1.6960e+0 (5.97e−1) −	1.0970e+0 (4.51e−1) −	6.5191e−1 (1.20e−1) −	7.1149e−1 (2.06e−1) −	2.3532e+1 (1.68e+1) −	5.8465e−1 (8.78e−2)
MaF8	5	2	6.8089e−1 (3.80e−1) −	4.5745e−1 (2.55e−1) −	9.5484e−1 (3.00e−1) −	5.5523e−1 (2.85e−1) −	4.3034e−1 (1.85e−1) −	2.7219e+2 (9.23e+2) −	3.8062e−1 (2.41e−1)
MaF9	5	2	8.4829e−1 (2.40e−1) −	1.3773e+0 (5.45e−1) −	1.2553e+0 (3.97e−1) −	4.1493e−1 (1.39e−1) =	3.7120e−1 (1.74e−1) +	5.3674e+2 (7.04e+2) −	4.0437e−1 (1.46e−1)
MaF10	5	14	1.4457e+0 (1.88e−1) =	1.2295e+0 (2.88e−1) +	1.4742e+0 (2.85e−1) =	1.3784e+0 (1.46e−1) =	1.2215e+0 (1.36e−1) +	2.8269e+0 (3.67e−1) −	1.4673e+0 (1.65e−1)
MaF11	5	14	1.0102e+0 (1.91e−1) =	1.4313e+0 (5.60e−1) −	1.1575e+0 (5.13e−1) =	9.0754e−1 (1.93e−1) +	9.0785e−1 (1.35e−1) +	9.6524e−1 (7.64e−2) =	1.0014e+0 (1.30e−1)
MaF12	5	14	2.0634e+0 (1.58e−1) −	2.4670e+0 (4.98e−1) −	2.0958e+0 (8.27e−2) −	1.9834e+0 (7.27e−2) −	2.1309e+0 (8.45e−2) −	2.1259e+0 (9.36e−2) −	1.8827e+0 (2.18e−2)
MaF13	5	5	5.5035e−1 (6.55e−2) −	4.6886e−1 (1.04e−1) −	6.4731e−1 (1.15e−1) −	4.0306e−1 (1.06e−1) −	2.5871e−1 (5.11e−2) +	7.4384e+0 (1.40e+1) −	3.5086e−1 (6.71e−2)
MaF14	5	100	8.8308e+1 (3.70e+2) +	7.3592e+0 (8.54e+0) +	7.6289e+0 (5.41e+0) +	1.2092e+1 (6.47e+0) +	4.3310e+0 (1.61e+0) +	3.1880e+0 (2.20e+0) +	2.9272e+2 (6.08e+2)
MaF15	5	100	6.0082e+0 (3.83e+0) +	9.9821e−1 (1.29e−1) +	1.2871e+0 (1.29e−1) +	5.1637e+0 (1.26e+0) +	6.5077e−1 (9.71e−2) +	2.2212e+0 (2.25e−1) +	1.4878e+1 (3.99e+0)
WFG1	5	14	1.3963e+0 (1.53e−1) =	1.2700e+0 (2.88e−1) +	1.4448e+0 (2.51e−1) =	1.3404e+0 (1.31e−1) +	1.2384e+0 (1.18e−1) +	2.7972e+0 (3.37e−1) −	1.4473e+0 (1.60e−1)
WFG2	5	14	1.0422e+0 (2.18e−1) =	1.3812e+0 (4.32e−1) −	1.0477e+0 (2.85e−1) =	9.9319e−1 (2.59e−1) =	9.7423e−1 (1.88e−1) =	9.3461e−1 (8.14e−2) =	1.0226e+0 (1.93e−1)
WFG3	5	14	1.1691e+0 (3.52e−1) −	1.9715e+0 (8.63e−1) −	1.9600e−1 (3.93e−2) +	8.3107e−1 (1.05e−1) −	9.4814e−1 (3.83e−1) −	1.6510e+0 (1.19e−1) −	7.6246e−1 (1.12e−1)
WFG4	5	14	2.0940e+0 (1.17e−1) −	2.8033e+0 (3.27e−1) −	2.3104e+0 (7.17e−2) −	1.9709e+0 (4.40e−2) +	2.2555e+0 (6.05e−2) −	2.3368e+0 (2.13e−1) −	2.0016e+0 (4.32e−2)
WFG5	5	14	2.0378e+0 (1.85e−2) −	2.3948e+0 (2.05e−1) −	2.2496e+0 (6.81e−2) −	2.0279e+0 (4.71e−2) −	2.2065e+0 (5.95e−2) −	2.0573e+0 (3.41e−2) −	1.9563e+0 (3.02e−2)
WFG6	5	14	2.1405e+0 (3.03e−1) −	2.4635e+0 (2.97e−1) −	2.3523e+0 (2.42e−1) −	1.9696e+0 (4.47e−2) +	2.3411e+0 (1.18e−1) −	2.1475e+0 (7.87e−2) −	2.0039e+0 (2.63e−2)
WFG7	5	14	2.1124e+0 (3.30e−2) −	2.4850e+0 (2.17e−1) −	2.4014e+0 (1.60e−1) −	2.0069e+0 (5.43e−2) =	2.2917e+0 (6.61e−2) −	2.1799e+0 (5.80e−2) −	2.0106e+0 (2.69e−2)
WFG8	5	14	2.0904e+0 (1.14e−1) −	2.5381e+0 (3.71e−1) −	2.2273e+0 (1.05e−1) −	2.0079e+0 (4.70e−2) −	2.2028e+0 (7.91e−2) −	2.3849e+0 (6.87e−2) −	1.9581e+0 (1.99e−2)
WFG9	5	14	2.0349e+0 (4.59e−2) −	2.5609e+0 (4.87e−1) −	2.0826e+0 (8.61e−2) −	1.9836e+0 (6.37e−2) −	2.1527e+0 (7.79e−2) −	2.1279e+0 (9.86e−2) −	1.8912e+0 (2.45e−2)
+/−/≈			5/15/4	6/18/0	5/14/5	9/10/5	9/13/2	3/19/2	

M is the number of objectives, and D is the dimensions of decision variables. Each row lists the IGD value(the value in parentheses is the standard deviation) of the algorithm under the problem, and the best value is highlighted in blue. The symbols "+", "−", and "≈" in the Wilcoxon test indicated that the algorithm performed better, worse, and similarly compared to $\mu MOGAIF$, respectively.

Table 7
Comparisons of average spread values of $\mu MOGAIF$ with NSGA-III, GrEA, IBEA, Two_Arch2, SPEA2+SDE, MOCPSO on 5-objective problems run independently for 30 times.

Problem	M	D	NSGA-III	GrEA	IBEA	Two_Arch2	SPEA2+SDE	MOCPSO	μMOGAIF
MaF1	5	14	9.9643e−1 (6.62e−2) −	7.0124e−1 (7.45e−2) −	6.6866e−1 (5.08e−2) −	4.3273e−1 (7.37e−2) −	3.2179e−1 (9.75e−2) −	-2.3757e+0 (2.11e+1) +	1.6326e−1 (5.19e−2)
MaF2	5	14	1.0166e+0 (1.25e−1) −	8.5677e−1 (3.70e−2) −	3.5957e−1 (7.77e−2) −	5.9465e−1 (9.55e−2) −	4.0644e−1 (1.20e−1) −	3.6649e−1 (3.35e−2) −	2.1412e−1 (1.74e−2)
MaF3	5	14	2.2943e+0 (1.86e−1) −	1.6878e+0 (7.43e−1) −	1.0001e+0 (2.29e−4) =	1.5270e+0 (4.19e−1) −	6.4156e−1 (1.78e−1) +	1.3179e+0 (1.78e+0) −	1.0684e+0 (5.64e−1)
MaF4	5	14	1.0490e+0 (3.01e−1) −	9.2613e−1 (4.96e−2) =	1.0190e+0 (4.21e−2) −	7.0890e−1 (9.97e−2) =	8.4669e−1 (1.51e−1) =	1.2964e+0 (3.71e−1) −	8.7942e−1 (5.11e−1)
MaF5	5	14	8.8735e−1 (3.22e−1) −	8.6334e−1 (2.11e−1) −	7.5236e−1 (1.94e−1) −	7.1402e−1 (1.50e−1) −	7.6559e−1 (2.29e−1) −	1.1579e+0 (1.26e−1) −	2.9556e−1 (1.72e−1)
MaF6	5	14	1.1783e+0 (2.00e−1) −	1.7797e+0 (1.54e−1) −	1.3792e+0 (2.61e−1) −	7.2051e−1 (1.72e−1) −	8.2416e−1 (1.36e−1) −	3.1241e−1 (2.07e+1) −	1.1710e−1 (3.45e−2)
MaF7	5	24	8.0832e−1 (1.75e−1) −	8.3596e−1 (6.48e−2) −	6.3879e−1 (1.07e−1) −	5.0341e−1 (9.82e−2) −	5.7721e−1 (6.17e−2) −	1.4141e+0 (3.72e−1) −	2.2082e−1 (6.49e−2)
MaF8	5	2	1.0836e+0 (7.99e−2) −	8.6902e−1 (1.58e−1) −	1.1416e+0 (2.38e−1) −	5.8121e−1 (2.28e−1) −	6.9509e−1 (1.32e−1) −	1.9924e+0 (1.08e+0) −	3.5407e−1 (1.92e−1)
MaF9	5	2	1.9953e+0 (5.24e−1) −	1.9916e+0 (6.17e−1) −	1.0771e+0 (1.59e−1) −	1.5687e+0 (6.60e−1) −	8.1594e−1 (3.98e−1) =	1.7354e+0 (0.00e+0) =	6.2066e−1 (2.39e−1)
MaF10	5	14	9.7867e−1 (1.88e−1) −	9.5632e−1 (9.77e−2) −	9.1553e−1 (5.98e−2) −	7.7060e−1 (1.02e−1) −	8.7093e−1 (6.75e−2) −	4.7922e+1 (2.31e+2) −	4.6782e−1 (1.23e−1)
MaF11	5	14	9.2068e−1 (7.00e−2) −	9.0310e−1 (7.23e−2) −	9.3398e−1 (5.26e−2) −	6.6411e−1 (1.28e−1) −	8.9655e−1 (5.36e−2) −	9.2539e−1 (7.35e−2) −	3.4267e−1 (6.90e−2)
MaF12	5	14	5.5617e−1 (8.06e−2) −	5.5938e−1 (1.45e−1) −	3.6381e−1 (6.66e−2) −	5.2944e−1 (1.59e−1) −	4.1682e−1 (7.39e−2) −	6.2986e−1 (6.69e−2) −	1.5611e−1 (2.77e−2)
MaF13	5	5	2.2900e+0 (2.93e−1) =	1.2880e+0 (2.82e−1) +	1.4593e+4 (7.99e+4) =	1.8625e+0 (6.14e−1) =	5.7779e−1 (1.72e−1) +	1.1606e+0 (4.91e+0) +	1.9924e+0 (6.00e−1)
MaF14	5	100	1.4191e+0 (3.21e−1) −	1.3077e+0 (2.84e−1) −	1.6377e+0 (4.93e−1) −	1.0452e+0 (3.46e−1) −	1.6383e+0 (4.24e−1) −	1.9004e+0 (1.61e+0) −	2.7260e−1 (1.93e−1)
MaF15	5	100	9.9705e−1 (9.07e−2) −	9.7798e−1 (1.35e−1) −	9.0685e−1 (1.49e−1) −	5.6528e−1 (1.06e−1) −	8.8518e−1 (2.89e−1) −	1.1378e+0 (1.81e−1) −	4.4603e−1 (7.34e−2)
WFG1	5	14	9.7815e−1 (2.08e−1) −	9.7118e−1 (1.04e−1) −	9.3858e−1 (7.36e−2) −	7.9743e−1 (9.67e−2) −	8.3494e−1 (8.17e−2) −	7.2734e−1 (1.84e+0) −	4.4124e−1 (7.88e−2)
WFG2	5	14	9.0368e−1 (1.22e−1) −	8.9332e−1 (6.32e−2) −	9.1768e−1 (6.52e−2) −	6.5017e−1 (1.10e−1) −	9.0978e−1 (5.61e−2) −	8.9180e−1 (4.86e−2) −	3.3288e−1 (7.53e−2)
WFG3	5	14	1.0257e+0 (1.32e−1) −	1.0535e+0 (4.73e−2) −	5.0306e−1 (1.82e−1) −	4.0489e−1 (7.55e−2) −	5.8795e−1 (5.89e−2) −	6.9063e−1 (7.22e−2) −	2.1492e−1 (4.88e−2)
WFG4	5	14	4.7776e−1 (2.09e−1) −	6.5111e−1 (1.44e−1) −	3.5135e−1 (8.90e−2) −	4.8734e−1 (1.74e−1) −	4.4403e−1 (8.66e−2) −	5.5242e−1 (6.71e−2) −	1.3119e−1 (4.38e−2)
WFG5	5	14	4.3536e−1 (5.69e−2) −	5.4117e−1 (1.39e−1) −	4.0283e−1 (8.71e−2) −	6.7912e−1 (1.29e−1) −	4.1349e−1 (7.60e−2) −	6.2091e−1 (4.27e−2) −	1.3088e−1 (2.92e−2)
WFG6	5	14	5.0014e−1 (1.59e−1) −	5.5384e−1 (1.10e−1) −	4.6766e−1 (1.21e−1) −	3.9808e−1 (1.24e−1) −	4.9791e−1 (9.91e−2) −	4.8454e−1 (3.96e−2) −	1.3783e−1 (3.39e−2)
WFG7	5	14	4.5352e−1 (6.57e−2) −	5.2159e−1 (1.07e−1) −	4.3287e−1 (1.00e−1) −	4.0222e−1 (1.26e−1) −	4.8603e−1 (9.04e−2) −	5.3603e−1 (6.39e−2) −	1.3510e−1 (3.50e−2)
WFG8	5	14	5.8437e−1 (1.29e−1) −	5.3863e−1 (1.96e−1) −	3.6151e−1 (7.20e−2) −	3.3158e−1 (1.12e−1) −	4.2347e−1 (7.99e−2) −	5.5585e−1 (8.00e−2) −	1.5099e−1 (3.65e−2)
WFG9	5	14	5.3318e−1 (8.79e−2) −	6.3129e−1 (2.19e−1) −	3.6960e−1 (7.96e−2) −	5.4034e−1 (1.51e−1) −	4.1567e−1 (5.99e−2) −	6.2775e−1 (5.74e−2) −	1.4616e−1 (3.35e−2)
+/−/≈			0/23/1	1/22/1	0/22/2	0/22/2	2/20/2	2/21/1	

M is the number of objectives, and D is the dimensions of decision variables. Each row lists the Spread value (the value in parentheses is the standard deviation) of the algorithm under the problem, and the best value is highlighted in blue. The symbols "+", "−", and "≈" in the Wilcoxon test indicated that the algorithm performed better, worse, and similarly compared to $\mu MOGAIF$, respectively.

Table 8
Comparisons of average IGD values of AMGA, AMGA2, μMMOGA, μMMOPSO, μMMABC, μMOEA, μMOGAIF on 5-objective problems run independently 30 times.

Problem	M	D	AMGA	AMGA2	μMMOGA	μMMOPSO	μMMABC	μMOEA	μMOGAIF
MaF1	5	14	2.8982e−1 (1.48e−2) −	3.6817e−1 (3.38e−2) −	2.9002e−1 (1.57e−2) −	2.8554e−1 (2.05e−2) −	3.5168e−1 (2.69e−2) −	2.9538e−1 (1.73e−2) −	2.5844e−1 (1.03e−2)
MaF2	5	14	2.1290e−1 (1.28e−2) −	2.3046e−1 (1.14e−2) −	2.3376e−1 (1.17e−2) −	2.3187e−1 (1.69e−2) −	2.3131e−1 (1.50e−2) −	2.3244e−1 (3.96e−3) −	1.9671e−1 (4.15e−3)
MaF3	5	14	7.2481e+3 (4.38e+3) +	2.5912e+6 (1.40e+7) −	4.8883e+3 (4.16e+3) +	2.8780e+4 (1.13e+4) +	8.8692e+3 (6.77e+3) +	5.4415e+2 (4.58e+2) +	1.2726e+6 (2.94e+6)
MaF4	5	14	5.9759e+2 (2.69e+2) −	1.7662e+3 (5.58e+2) −	3.0329e+2 (1.86e+2) =	1.0440e+3 (4.87e+2) −	2.9327e+2 (4.49e+2) +	3.4300e+2 (1.73e+2) =	3.3015e+2 (2.03e+2)
MaF5	5	14	1.7712e+1 (8.48e+0) −	1.1360e+1 (5.47e+0) −	7.8808e+0 (7.56e+0) =	4.1373e+0 (2.28e−1) =	4.7751e+0 (2.90e−1) +	1.1013e+1 (7.71e+0) −	5.0283e+0 (1.63e+0)
MaF6	5	14	4.2477e−2 (4.61e−3) −	1.2199e+0 (1.41e+0) −	3.6223e−2 (5.87e−3) −	3.0893e−2 (3.66e−3) −	6.9564e−2 (1.48e−2) −	6.6963e−2 (5.55e−2) −	2.0950e−2 (2.14e−4)
MaF7	5	24	7.8288e−1 (2.60e−1) −	1.1206e+0 (3.93e−1) −	9.0662e−1 (1.90e−1) −	6.7693e−1 (1.56e−1) −	6.5575e−1 (3.38e−2) −	1.1411e+0 (1.08e−1) −	5.8465e−1 (8.78e−2)
MaF8	5	2	6.6926e−1 (4.31e−1) −	4.3984e−1 (2.79e−1) −	4.8859e−1 (1.34e−1) −	4.1404e−1 (5.22e−2) −	3.6129e−1 (4.27e−2) +	1.0089e+0 (8.66e−1) −	3.8062e−1 (2.41e−1)
MaF9	5	2	1.3402e+2 (1.58e+2) −	2.3700e+1 (2.38e+1) −	1.6741e+1 (2.06e+1) −	2.0117e+0 (1.98e+0) −	1.5718e+0 (7.28e−1) −	6.9415e−1 (4.52e−1) −	4.0437e−1 (1.46e−1)
MaF10	5	14	2.0086e+0 (1.26e−1) −	2.2447e+0 (5.32e−2) −	1.8454e+0 (1.58e−1) −	1.8577e+0 (1.53e−1) −	2.9158e+0 (1.74e−1) −	1.5731e+0 (1.65e−1) −	1.4673e+0 (1.65e−1)
MaF11	5	14	1.5135e+0 (1.62e−1) −	1.3420e+0 (1.43e−1) −	1.5334e+0 (1.75e−1) −	1.3868e+0 (1.74e−1) −	9.7360e−1 (7.77e−2) =	1.0503e+0 (8.40e−2) −	1.0014e+0 (1.30e−1)
MaF12	5	14	2.2290e+0 (1.26e−1) −	2.2263e+0 (7.06e−2) −	2.2969e+0 (1.57e−1) −	2.2402e+0 (1.09e−1) −	1.9750e+0 (4.01e−2) −	2.8034e+0 (1.16e−1) −	1.8827e+0 (2.18e−2)
MaF13	5	5	6.5427e−1 (1.36e−1) −	4.3975e−1 (9.50e−2) −	6.5602e−1 (1.62e−1) −	7.3740e−1 (1.99e−1) −	6.4310e−1 (8.83e−2) −	4.5495e−1 (8.59e−2) −	3.5086e−1 (6.71e−2)
MaF14	5	100	6.6138e+0 (6.54e+0) +	6.3728e+0 (4.45e+0) +	2.2097e+1 (7.74e+0) =	9.5883e−1 (0.00e+0) +	9.5764e−1 (2.70e−3) +	5.8520e+0 (2.47e+0) +	2.9272e+2 (6.08e+2)
MaF15	5	100	6.4161e+0 (4.94e+0) +	1.3557e+1 (8.42e+0) =	1.8381e+1 (8.75e+0) −	1.4286e+1 (6.69e+0) =	9.9915e−1 (6.54e−1) +	8.6912e−1 (6.50e−2) +	1.4878e+1 (3.99e+0)
WFG1	5	14	2.0333e+0 (1.36e−1) −	2.2483e+0 (5.54e−2) −	1.9459e+0 (1.20e−1) −	1.8378e+0 (1.31e−1) −	2.9477e+0 (1.25e−1) −	1.7596e+0 (1.27e−1) −	1.4473e+0 (1.60e−1)
WFG2	5	14	1.4728e+0 (1.34e−1) −	1.4090e+0 (1.85e−1) −	1.4972e+0 (2.18e−1) −	1.4105e+0 (1.80e−1) −	9.8151e−1 (4.90e−2) =	1.0511e+0 (7.85e−2) −	1.0226e+0 (1.93e−1)
WFG3	5	14	1.1709e+0 (2.30e−1) −	9.8986e−1 (1.32e−1) −	1.3586e+0 (2.66e−1) −	1.2037e+0 (2.35e−1) −	1.5568e+0 (1.64e−1) −	1.8803e+0 (9.73e−2) −	7.6246e−1 (1.12e−1)
WFG4	5	14	2.1294e+0 (9.29e−2) −	2.1285e+0 (1.12e−1) −	2.1022e+0 (5.65e−2) −	2.1336e+0 (7.84e−2) −	2.0664e+0 (3.00e−2) −	3.0348e+0 (1.23e−1) −	2.0016e+0 (4.32e−2)
WFG5	5	14	2.2673e+0 (6.90e−2) −	2.0560e+0 (7.05e−2) −	2.1349e+0 (7.23e−2) −	2.2049e+0 (7.87e−2) −	2.0022e+0 (2.15e−2) −	2.8761e+0 (1.09e−1) −	1.9563e+0 (3.02e−2)
WFG6	5	14	2.2029e+0 (8.37e−2) −	2.3542e+0 (9.08e−2) −	2.1039e+0 (8.81e−2) −	2.3285e+0 (1.90e−1) −	2.1072e+0 (4.81e−2) −	2.6529e+0 (1.51e−1) −	2.0039e+0 (2.63e−2)
WFG7	5	14	2.2900e+0 (1.22e−1) −	2.2895e+0 (1.38e−1) −	2.1755e+0 (7.01e−2) −	2.1416e+0 (8.03e−2) −	2.1237e+0 (3.92e−2) −	2.6923e+0 (1.35e−1) −	2.0106e+0 (2.69e−2)
WFG8	5	14	2.2264e+0 (5.56e−2) −	2.2952e+0 (6.73e−2) −	2.1466e+0 (6.77e−2) −	2.2243e+0 (8.75e−2) −	2.0877e+0 (2.97e−2) −	2.9582e+0 (8.77e−2) −	1.9581e+0 (1.99e−2)
WFG9	5	14	2.2817e+0 (1.58e−1) −	2.2219e+0 (7.76e−2) −	2.3490e+0 (5.01e−1) −	2.2812e+0 (9.62e−2) −	1.9944e+0 (4.77e−2) −	2.7847e+0 (9.71e−2) −	1.8912e+0 (2.45e−2)
+/−/≈			3/21/0	1/22/1	1/20/3	2/20/2	6/16/2	3/20/1	

M is the number of objectives, and D is the dimensions of decision variables. Each row lists the IGD value(the value in parentheses is the standard deviation) of the algorithm under the problem, and the best value is highlighted in blue. The symbols "+", "−", and "≈" in the Wilcoxon test indicated that the algorithm performed better, worse, and similarly compared to $\mu MOGAIF$, respectively.

Table 9
Comparisons of average spread values of AMGA, AMGA2, μMMOGA, μMMOPSO, μMMABC, μMOGAIF on 5-objective problems run independently 30 times.

Problem	M	D	AMGA	AMGA2	μMMOGA	μMMOPSO	μMMABC	μMOEA	μMOGAIF
MaF1	5	14	5.9822e−1 (1.19e−1) −	3.9777e−1 (9.24e−2) −	4.7680e−1 (1.35e−1) −	4.7961e−1 (1.08e−1) −	1.1075e+0 (1.48e−1) −	1.1480e+0 (5.50e−2) −	1.6326e−1 (5.19e−2)
MaF2	5	14	7.6464e−1 (1.10e−1) −	6.0556e−1 (7.74e−2) −	7.4809e−1 (1.18e−1) −	8.1318e−1 (1.42e−1) −	9.0885e−1 (1.48e−1) −	1.2114e+0 (1.84e−2) −	2.1412e−1 (1.74e−2)
MaF3	5	14	2.1819e+0 (1.76e−1) −	1.8803e+0 (2.35e−1) −	2.1805e+0 (2.01e−1) −	2.0353e+0 (1.86e−1) −	2.2547e+0 (2.68e−1) −	1.7365e+0 (3.49e−1) −	1.0684e+0 (5.64e−1)
MaF4	5	14	8.2263e−1 (9.13e−2) =	6.6638e−1 (1.05e−1) =	7.5913e−1 (1.20e−1) =	7.3717e−1 (6.61e−2) =	1.9816e+0 (3.86e−1) −	1.3087e+0 (3.67e−2) −	8.7942e−1 (5.11e−1)
MaF5	5	14	8.0071e−1 (1.93e−1) −	9.1399e−1 (4.56e−1) −	8.5598e−1 (1.61e−1) −	7.5297e−1 (1.22e−1) −	6.7348e−1 (9.50e−2) −	1.1001e+0 (3.58e−1) −	2.9556e−1 (1.72e−1)
MaF6	5	14	1.3129e+0 (1.07e−1) −	2.2280e+0 (5.29e+0) −	1.0705e+0 (2.49e−1) −	7.7370e−1 (1.46e−1) −	1.2105e+0 (2.16e−1) −	7.5701e−1 (4.01e−1) −	1.1710e−1 (3.45e−2)
MaF7	5	24	7.0672e−1 (1.18e−1) −	5.9346e−1 (5.99e−2) −	7.1108e−1 (1.24e−1) −	6.4683e−1 (1.01e−1) −	7.7121e−1 (2.30e−1) −	1.4138e+0 (9.18e−2) −	2.2082e−1 (6.49e−2)
MaF8	5	2	1.1530e+0 (1.80e+0) −	4.7634e−1 (1.68e−1) −	8.0990e−1 (1.85e−1) −	5.4940e−1 (1.34e−1) −	1.4335e+0 (1.85e−1) −	1.0531e+0 (7.15e−2) −	3.5407e−1 (1.92e−1)
MaF9	5	2	1.4570e+0 (2.68e−1) −	1.5105e+0 (2.52e−1) −	1.6212e+0 (2.80e−1) −	1.9116e+0 (4.17e−1) −	2.1330e+0 (4.68e−1) −	1.3626e+0 (3.15e−1) −	6.2066e−1 (2.39e−1)
MaF10	5	14	9.7699e−1 (1.31e−1) −	8.7715e−1 (9.35e−2) −	8.3613e−1 (1.17e−1) −	7.3084e−1 (1.16e−1) −	1.4446e+0 (2.49e−1) −	1.3879e+0 (1.44e−1) −	4.6782e−1 (1.23e−1)
MaF11	5	14	8.7679e−1 (1.67e−1) −	6.6338e−1 (9.72e−2) −	6.9626e−1 (1.64e−1) −	8.2749e−1 (1.40e−1) −	7.6401e−1 (9.84e−2) −	1.5567e+0 (1.31e−1) −	3.4267e−1 (6.90e−2)
MaF12	5	14	8.0803e−1 (1.24e−1) −	6.7532e−1 (8.24e−2) −	7.3271e−1 (1.45e−1) −	7.7970e−1 (1.38e−1) −	5.7877e−1 (7.87e−2) −	1.2453e+0 (1.65e−1) −	1.5611e−1 (2.77e−2)
MaF13	5	5	2.0470e+0 (2.66e−1) =	2.4242e+0 (1.31e−1) −	2.1358e+0 (2.27e−1) =	2.0016e+0 (3.13e−1) =	2.3621e+0 (2.91e−1) −	1.1614e+0 (6.38e−2) +	1.9924e+0 (6.00e−1)
MaF14	5	100	1.1312e+0 (2.51e−1) −	8.8402e−1 (2.40e−1) −	1.1817e+0 (2.38e−1) −	9.9701e−1 (1.69e−1) −	1.8434e+0 (5.76e−1) −	2.2744e+0 (4.24e−1) −	2.7260e−1 (1.93e−1)
MaF15	5	100	8.1717e−1 (1.71e−1) −	6.3789e−1 (7.95e−2) −	8.7715e−1 (1.07e−1) −	7.9401e−1 (8.34e−2) −	1.4679e+0 (2.78e−1) −	1.1906e+0 (9.78e−2) −	4.4603e−1 (7.34e−2)
WFG1	5	14	9.9356e−1 (1.19e−1) −	8.6843e−1 (8.13e−2) −	8.6362e−1 (8.42e−2) −	7.2108e−1 (1.21e−1) −	1.4806e+0 (2.35e−1) −	1.3643e+0 (1.29e−1) −	4.4124e−1 (7.88e−2)
WFG2	5	14	7.7907e−1 (1.59e−1) −	6.8267e−1 (1.20e−1) −	7.8498e−1 (1.39e−1) −	8.0715e−1 (1.89e−1) −	7.9194e−1 (5.87e−2) −	1.5322e+0 (1.10e−1) −	3.3288e−1 (7.53e−2)
WFG3	5	14	5.4807e−1 (1.32e−1) −	4.8864e−1 (9.05e−2) −	6.4703e−1 (1.10e−1) −	6.5756e−1 (1.77e−1) −	8.7495e−1 (1.53e−1) −	1.5825e+0 (1.02e−1) −	2.1492e−1 (4.88e−2)
WFG4	5	14	6.6093e−1 (1.71e−1) −	5.5580e−1 (1.22e−1) −	6.4866e−1 (1.52e−1) −	6.8806e−1 (1.61e−1) −	3.9476e−1 (3.80e−2) −	1.5634e+0 (1.90e−1) −	1.3119e−1 (4.38e−2)
WFG5	5	14	8.2623e−1 (9.17e−2) −	6.6205e−1 (8.78e−2) −	7.6988e−1 (1.54e−1) −	9.3760e−1 (1.04e−1) −	4.8639e−1 (4.89e−2) −	1.3146e+0 (1.29e−1) −	1.3088e−1 (2.92e−2)
WFG6	5	14	7.7782e−1 (1.45e−1) −	6.2418e−1 (1.36e−1) −	6.6284e−1 (1.93e−1) −	9.5687e−1 (3.68e−1) −	4.0922e−1 (6.82e−2) −	1.5576e+0 (1.59e−1) −	1.3783e−1 (3.39e−2)
WFG7	5	14	8.1722e−1 (1.36e−1) −	5.5117e−1 (7.53e−2) −	7.3051e−1 (1.79e−1) −	6.1813e−1 (1.71e−1) −	4.3682e−1 (5.80e−2) −	1.5891e+0 (1.57e−1) −	1.3510e−1 (3.50e−2)
WFG8	5	14	7.3754e−1 (1.44e−1) −	5.8119e−1 (7.05e−2) −	6.8220e−1 (1.71e−1) −	6.1222e−1 (1.55e−1) −	5.2482e−1 (6.67e−2) −	1.7073e+0 (5.49e−2) −	1.5099e−1 (3.65e−2)
WFG9	5	14	7.8575e−1 (1.46e−1) −	6.4372e−1 (7.47e−2) −	7.5043e−1 (1.26e−1) −	8.3730e−1 (1.06e−1) −	5.9731e−1 (9.88e−2) −	1.2568e+0 (1.55e−1) −	1.4616e−1 (3.35e−2)
+/−/≈			0/22/2	0/23/1	0/22/2	0/22/2	0/24/0	1/23/0	

M is the number of objectives, and D is the dimensions of decision variables. Each row lists the Spread value (the value in parentheses is the standard deviation) of the algorithm under the problem, and the best value is highlighted in blue. The symbols "+", "−", and "≈" in the Wilcoxon test indicated that the algorithm performed better, worse, and similarly compared to $\mu MOGAIF$, respectively.

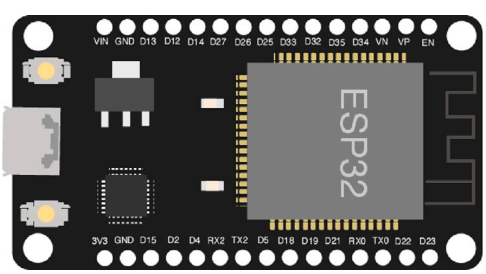


Fig. 8. Schematic diagram of low-power microprocessor ESP32-WROOM.

consumption, scalability, and adaptability. Fig. 8 shows an example image of the ESP32-WROOM.

The simulation experiment was conducted on DTLZ benchmark problems and compared the performance between MOEA/D, MOEA/D-DE, μMOPSO, μMOPSO, μMOEA, μGA and μMOGAIF on the ESP32-WROOM. The programming language used is MicroPython, a streamlined version of Python designed to run on microcontrollers with limited storage and computing resources. The parameter settings for the simulation experiment are as specified in Section 5.4. Tables 13 and 14 present the detailed average IGD and standard deviation.

The four manual testing questions selected have significant guidance for detecting algorithm performance. For the DTLZ1 problem with multiple locally optimal fronts in the search space, μMOGAIF demonstrated superior optimization ability compared to other algorithms, particularly outperforming μGA, and the standard deviation has also reached the lowest value. The results of the DTLZ2 and DTLZ3 problems, which aim to test the ability to search for the global optimal front, indicate that μMOGAIF outperformed the other six algorithms. The DTLZ4 problem is designed to evaluate the capability of algorithms to maintain a well-distributed solution set. μMOGAIF was not as effective as the traditional MOEA MOEA/D-DE, and the micro population MOEA μMOPSO, μMOEA. During the optimization of the DTLZ4 problem, the performance of the final solution set obtained by the algorithm depends on the distribution performance of the initial population. In addition, the optimization results of μMOGAIF on ESP32-WROOM are comparable to those of high-end computers and even better on the DTLZ2 problem.

6. Conclusion

In this paper, a micro multi-objective genetic algorithm with information fitting strategy (μMOGAIF) for low-power microprocessors is proposed. μMOGAIF promotes solution exploration at the Pareto front and consistently guides the population to converge towards superior fronts while preserving a uniform distribution. An information fitting strategy is enabled after generating a new offspring population, which provides population state information for the algorithm. In addition,

Table 10
Comparisons of average IGD values of $\mu MOGAIF$ with Variant 1, Variant 2, Variant 3 on DTLZ, MaF, WFG, and ZDT test suites run 30 times independently. The bottom part records the average ranking of these algorithms in the Friedman test.

Problem	M	D	Variant 1	Variant 2	Variant 3	μMOGAIF
DTLZ1	3	7	7.8039e−1 (7.53e−1)	4.9363e−1 (1.48e−1)	1.9062e−1 (2.40e−1)	1.3898e−1 (1.74e−1)
DTLZ2	3	12	1.3300e−1 (2.50e−3)	8.2722e−1 (1.32e−1)	1.3367e−1 (2.05e−3)	1.3274e−1 (2.30e−3)
DTLZ3	3	12	2.2692e+1 (1.17e+1)	5.6841e+0 (3.71e+0)	8.8462e+0 (6.45e+0)	1.1436e+1 (8.30e+0)
DTLZ4	3	12	7.2498e−1 (2.56e−1)	9.4593e−1 (2.28e−7)	7.0351e−1 (2.56e−1)	6.7262e−1 (3.10e−1)
DTLZ5	3	12	2.1846e−2 (4.37e−4)	7.4209e−1 (1.32e−7)	2.1843e−2 (4.26e−4)	2.1945e−2 (5.41e−4)
DTLZ6	3	12	2.2845e−2 (3.28e−3)	7.4393e−1 (3.74e−3)	2.1096e−2 (9.37e−4)	2.0931e−2 (1.95e−4)
DTLZ7	3	22	3.5313e−1 (2.30e−1)	1.3660e+0 (1.42e−1)	3.4068e−1 (2.22e−1)	3.4177e−1 (2.19e−1)
MaF1	3	12	1.0511e−1 (2.93e−3)	4.1770e−1 (5.42e−2)	1.0413e−1 (1.54e−3)	1.0215e−1 (2.53e−3)
MaF2	3	12	7.2588e−2 (1.60e−3)	4.8138e−1 (5.72e−3)	7.2060e−2 (1.64e−3)	7.1953e−2 (1.59e−3)
MaF3	3	12	3.8868e+4 (1.20e+5)	1.9222e+4 (8.66e+4)	3.0294e+3 (8.52e+3)	6.0973e+3 (1.81e+4)
MaF4	3	12	1.1187e+2 (5.01e+1)	3.7398e+1 (2.80e+1)	5.7940e+1 (3.19e+1)	5.8311e+1 (3.57e+1)
MaF5	3	12	2.8131e+0 (1.76e+0)	4.8885e+0 (3.19e−6)	3.0720e+0 (1.72e+0)	3.7393e+0 (1.51e+0)
MaF6	3	12	2.1537e−2 (4.83e−4)	7.4211e−1 (9.81e−6)	2.0938e−2 (1.94e−4)	2.0937e−2 (2.21e−4)
MaF7	3	22	3.0596e−1 (2.25e−1)	1.3502e+0 (1.31e−1)	3.0558e−1 (2.30e−1)	2.9880e−1 (2.33e−1)
WFG1	3	12	1.2714e+0 (1.13e−1)	2.3659e+0 (7.39e−1)	5.4888e−1 (1.09e−1)	5.3933e−1 (1.29e−1)
WFG2	3	12	3.9885e−1 (5.28e−2)	4.7854e+0 (2.66e−1)	3.9524e−1 (8.66e−2)	3.7938e−1 (6.93e−2)
WFG3	3	12	2.0215e−1 (2.59e−2)	3.1936e+0 (7.56e−2)	2.0279e−1 (3.02e−2)	1.7792e−1 (2.30e−2)
WFG4	3	12	5.1571e−1 (8.47e−3)	3.9672e+0 (1.32e−1)	5.1532e−1 (1.06e−2)	5.1679e−1 (1.10e−2)
WFG5	3	12	5.0049e−1 (1.03e−2)	3.7981e+0 (4.39e−1)	5.0206e−1 (1.11e−2)	5.0144e−1 (9.77e−3)
WFG6	3	12	5.3153e−1 (1.54e−2)	3.7890e+0 (5.93e−1)	5.1992e−1 (8.88e−3)	5.1707e−1 (8.25e−3)
WFG7	3	12	5.1939e−1 (1.46e−2)	3.9283e+0 (3.51e−1)	5.1910e−1 (8.57e−3)	5.1909e−1 (1.20e−2)
WFG8	3	12	5.4980e−1 (1.56e−2)	3.9407e+0 (2.30e−1)	5.4350e−1 (1.39e−2)	5.4191e−1 (1.00e−2)
WFG9	3	12	5.1878e−1 (2.26e−2)	3.6158e+0 (6.36e−1)	5.0217e−1 (2.08e−2)	5.1341e−1 (2.56e−2)
ZDT1	2	30	2.2903e−2 (4.20e−3)	6.9090e−1 (9.67e−2)	1.9917e−2 (5.77e−4)	1.9436e−2 (3.49e−4)
ZDT2	2	30	2.0625e−2 (1.85e−3)	6.0384e−1 (2.03e−2)	1.9908e−2 (3.15e−4)	1.9767e−2 (2.27e−4)
ZDT3	2	30	5.7371e−2 (4.48e−2)	7.0133e−1 (7.52e−2)	3.7995e−2 (2.21e−2)	4.4526e−2 (3.94e−2)
ZDT4	2	10	1.2460e−1 (7.44e−2)	8.2931e−1 (1.41e−1)	3.1168e−1 (1.02e−1)	2.6327e−1 (1.26e−1)
ZDT6	2	10	1.5910e−2 (1.50e−4)	1.5622e−1 (6.95e−2)	1.5635e−2 (2.44e−4)	1.5647e−2 (1.51e−4)
Average rank			4.23	6.46	2.80	2.55

M is the number of objectives, and D is the dimensions of decision variables. Each row lists the IGD value(the value in parentheses is the standard deviation) of the algorithm under the problem, and the best value is highlighted in blue.

Table 11
Comparisons of average IGD values of $\mu MOGAIF$ with Variant 4, Variant 5, Variant 6 on DTLZ, MaF, WFG, and ZDT test suites run 30 times independently. The bottom part records the average ranking of these algorithms in the Friedman test.

Problem	M	D	Variant 4	Variant 5	Variant 6	μMOGAIF
DTLZ1	3	7	1.0690e−1 (1.44e−1)	2.2844e−1 (3.51e−1)	5.0639e−1 (2.60e−1)	1.3898e−1 (1.74e−1)
DTLZ2	3	12	1.3388e−1 (2.27e−3)	1.3248e−1 (1.71e−3)	3.0435e−1 (4.53e−2)	1.3274e−1 (2.30e−3)
DTLZ3	3	12	5.3395e+0 (3.09e+0)	1.3017e+1 (7.88e+0)	1.0221e+1 (7.36e+0)	1.1436e+1 (8.30e+0)
DTLZ4	3	12	5.1189e−1 (3.07e−1)	6.1242e−1 (2.58e−1)	9.4259e−1 (2.11e−2)	6.7262e−1 (3.10e−1)
DTLZ5	3	12	2.1558e−2 (2.82e−4)	2.1780e−2 (4.31e−4)	1.3803e−1 (5.02e−2)	2.1945e−2 (5.41e−4)
DTLZ6	3	12	2.1582e−2 (2.98e−4)	2.1011e−2 (2.71e−4)	2.3108e−1 (6.80e−2)	2.0931e−2 (1.95e−4)
DTLZ7	3	22	3.6080e−1 (2.37e−1)	3.3931e−1 (2.55e−1)	1.1286e+0 (3.62e−1)	3.4177e−1 (2.19e−1)
MaF1	3	12	1.0518e−1 (2.65e−3)	1.0255e−1 (2.52e−3)	2.1737e−1 (4.47e−2)	1.0215e−1 (2.53e−3)
MaF2	3	12	7.2819e−2 (1.53e−3)	7.2366e−2 (1.25e−3)	1.0566e−1 (1.58e−2)	7.1953e−2 (1.59e−3)
MaF3	3	12	3.4838e+3 (7.41e+3)	8.6469e+3 (3.00e+4)	9.8167e+2 (2.42e+3)	6.0973e+3 (1.81e+4)
MaF4	3	12	2.4710e+1 (1.48e+1)	5.8143e+1 (3.08e+1)	3.8049e+1 (2.44e+1)	5.8311e+1 (3.57e+1)
MaF5	3	12	2.3180e+0 (1.68e+0)	3.0975e+0 (1.88e+0)	4.5101e+0 (9.39e−1)	3.7393e+0 (1.51e+0)
MaF6	3	12	2.1386e−2 (3.67e−4)	2.0922e−2 (2.54e−4)	3.3554e−1 (8.50e−2)	2.0937e−2 (2.21e−4)
MaF7	3	22	3.1376e−1 (2.18e−1)	3.8268e−1 (2.57e−1)	1.1742e+0 (3.64e−1)	2.9880e−1 (2.33e−1)
WFG1	3	12	5.2061e−1 (1.01e−1)	5.5824e−1 (1.44e−1)	2.3168e+0 (1.26e−1)	5.3933e−1 (1.29e−1)
WFG2	3	12	4.2629e−1 (7.88e−2)	4.1083e−1 (9.12e−2)	1.0799e+0 (4.24e−1)	3.7938e−1 (6.93e−2)
WFG3	3	12	1.7800e−1 (1.69e−2)	1.9994e−1 (3.00e−2)	3.2893e−1 (8.07e−2)	1.7792e−1 (2.30e−2)
WFG4	3	12	5.2312e−1 (1.36e−2)	5.1577e−1 (1.45e−2)	1.1814e+0 (3.65e−1)	5.1679e−1 (1.10e−2)
WFG5	3	12	5.1031e−1 (1.08e−2)	5.0219e−1 (8.80e−3)	9.1310e−1 (1.43e−1)	5.0144e−1 (9.77e−3)
WFG6	3	12	5.2660e−1 (1.24e−2)	5.1724e−1 (7.78e−3)	9.5344e−1 (1.95e−1)	5.1707e−1 (8.25e−3)
WFG7	3	12	5.2544e−1 (1.57e−2)	5.1944e−1 (9.14e−3)	1.0686e+0 (2.47e−1)	5.1909e−1 (1.20e−2)
WFG8	3	12	5.4550e−1 (1.23e−2)	5.4421e−1 (1.05e−2)	9.2423e−1 (1.75e−1)	5.4191e−1 (1.00e−2)
WFG9	3	12	5.0982e−1 (1.85e−2)	5.0848e−1 (2.39e−2)	1.1006e+0 (2.65e−1)	5.1341e−1 (2.56e−2)
ZDT1	2	30	2.0384e−2 (5.97e−4)	1.9765e−2 (4.28e−4)	2.1202e−1 (8.27e−2)	1.9436e−2 (3.49e−4)
ZDT2	2	30	2.0476e−2 (4.98e−4)	1.9833e−2 (2.67e−4)	5.8631e−1 (8.39e−2)	1.9767e−2 (2.27e−4)
ZDT3	2	30	5.5174e−2 (4.86e−2)	4.3806e−2 (3.11e−2)	2.0951e−1 (6.87e−2)	4.4526e−2 (3.94e−2)
ZDT4	2	10	2.2417e−1 (1.04e−1)	2.8467e−1 (9.83e−2)	8.7523e−1 (1.08e−1)	2.6327e−1 (1.26e−1)
ZDT6	2	10	1.6111e−2 (2.75e−4)	1.5596e−2 (1.21e−4)	4.9847e−1 (6.91e−2)	1.5647e−2 (1.51e−4)
Average rank			3.30	2.93	5.71	2.55

M is the number of objectives, and D is the dimensions of decision variables. Each row lists the IGD value(the value in parentheses is the standard deviation) of the algorithm under the problem, and the best value is highlighted in blue.

utilizing two types of parents enhances the exploration of individuals during adaptive mating selection. While the dual archives utilized in μMOGAIF is not entirely novel and has been explored in previous works, our approach refines this concept by employing a compensatory archive to enhance convergence and diversity balance. The dual archives work in tandem, with one archive focusing on enhancing

Table 12
Comparisons of average IGD values of $\mu MOGAIF$ with different crossover operators on DTLZ, WFG, and ZDT test suites run 30 times independently. The bottom part records the average ranking of these algorithms in the Friedman test.

Problem	M	D	μMOGAIF-1	μMOGAIF-2	μMOGAIF-3	μMOGAIF
DTLZ1	3	7	1.8528e−1 (1.60e−1)	1.8717e−1 (1.51e−1)	1.7064e−1 (1.92e−1)	1.7000e−1 (2.32e−1)
DTLZ2	3	12	1.3251e−1 (2.40e−3)	1.3289e−1 (2.54e−3)	1.3354e−1 (2.61e−3)	1.3312e−1 (1.75e−3)
DTLZ3	3	12	1.2034e+1 (6.86e+0)	8.5164e+0 (5.07e+0)	9.3838e+0 (5.86e+0)	1.3317e+1 (9.39e+0)
DTLZ4	3	12	7.5755e−1 (2.31e−1)	7.4372e−1 (2.96e−1)	7.7056e−1 (2.54e−1)	6.4923e−1 (3.00e−1)
DTLZ5	3	12	2.1871e−2 (8.29e−4)	2.1708e−2 (6.18e−4)	2.1998e−2 (7.18e−4)	2.1832e−2 (4.80e−4)
DTLZ6	3	12	5.1545e−2 (1.67e−1)	3.6208e−2 (4.50e−2)	2.1101e−2 (4.11e−4)	2.0965e−2 (2.03e−4)
DTLZ7	3	22	5.0911e−1 (2.45e−1)	6.1163e−1 (2.41e−1)	5.5287e−1 (2.59e−1)	3.5200e−1 (2.26e−1)
WFG1	3	12	6.7536e−1 (1.15e−1)	7.5319e−1 (1.89e−1)	6.2524e−1 (1.01e−1)	5.4992e−1 (1.06e−1)
WFG2	3	12	4.5201e−1 (7.11e−2)	4.5192e−1 (1.89e−1)	4.4864e−1 (1.16e−1)	4.0391e−1 (9.09e−2)
WFG3	3	12	1.8876e−1 (2.53e−2)	1.7950e−1 (3.20e−2)	1.9878e−1 (2.84e−2)	1.8650e−1 (3.12e−2)
WFG4	3	12	5.1832e−1 (1.28e−2)	5.1134e−1 (1.07e−2)	5.1703e−1 (1.13e−2)	5.1077e−1 (1.03e−2)
WFG5	3	12	4.9746e−1 (9.24e−3)	4.9945e−1 (7.85e−3)	5.0016e−1 (8.12e−3)	5.0381e−1 (1.17e−2)
WFG6	3	12	5.2286e−1 (9.89e−3)	5.2190e−1 (9.52e−3)	5.2797e−1 (9.53e−3)	5.1814e−1 (9.37e−3)
WFG7	3	12	5.1537e−1 (1.04e−2)	5.1837e−1 (9.30e−3)	5.2226e−1 (1.17e−2)	5.1851e−1 (9.89e−3)
WFG8	3	12	5.6198e−1 (1.28e−2)	5.5580e−1 (1.05e−2)	5.5844e−1 (1.28e−2)	5.4233e−1 (1.10e−2)
WFG9	3	12	5.0595e−1 (2.83e−2)	5.1377e−1 (2.81e−2)	5.0338e−1 (2.53e−2)	5.0103e−1 (1.91e−2)
ZDT1	2	30	2.0186e−2 (5.60e−4)	2.0491e−2 (1.33e−3)	2.0063e−2 (3.23e−4)	1.9644e−2 (3.49e−4)
ZDT2	2	30	5.3008e−2 (6.40e−2)	1.1661e−1 (9.78e−2)	2.0870e−2 (3.44e−3)	1.9926e−2 (7.65e−4)
ZDT3	2	30	5.6048e−2 (5.01e−2)	6.2449e−2 (6.08e−2)	5.3936e−2 (5.05e−2)	5.9961e−2 (5.05e−2)
ZDT4	2	10	2.2877e−1 (1.20e−1)	2.9592e−1 (1.34e−1)	1.4384e−1 (1.08e−1)	2.7912e−1 (1.08e−1)
ZDT6	2	10	1.5757e−2 (1.58e−4)	1.6537e−2 (8.02e−4)	1.5838e−2 (2.66e−4)	1.5654e−2 (1.79e−4)
Average rank			2.76	2.81	2.69	1.74

M is the number of objectives, and D is the dimensions of decision variables. Each row lists the IGD value(the value in parentheses is the standard deviation) of the algorithm under the problem, and the best value is highlighted in blue.

Table 13
Comparisons of average IGD values of μMOGAIF with MOEA/D , MOEA/D-DE, and MOPSO on DTLZ problems.

Problem	M	D	MOEA/D	MOEA/D-DE	MOPSO	μMOGAIF
DTLZ1	3	7	9.0127e−1 (8.53e−1) −	2.4524e+0 (3.15e+0) −	5.1894e+0 (2.92e+0) −	1.9039e−1 (1.75e−1)
DTLZ2	3	12	1.5903e−1 (9.94e−4) −	3.7318e−1 (3.85e−2) −	5.8027e−1 (4.75e−2) −	1.3345e−1 (1.96e−3)
DTLZ3	3	12	2.1323e+1 (9.87e+0) −	4.7219e+1 (4.24e+1) −	4.9239e+1 (4.07e+1) −	1.2170e+1 (9.42e+0)
DTLZ4	3	12	6.0207e−1 (2.25e−1) =	4.2106e−1 (1.18e−1) +	7.3316e−1 (1.79e−1) −	7.0366e−1 (2.51e−1)
+/−/≈			0/3/1	1/3/0	0/4/0	
Average ranking			3.125	3.75	5.5	2

Table 14
Comparisons of average IGD values of μMOGAIF with μMOPSO, μMOEA, and μGA on DTLZ problems.

Pro	M	D	μMOPSO	μMOEA	μGA	μMOGAIF
DTLZ1	3	7	1.3624e+1 (8.18e+0) −	8.7577e−1 (1.18e+0) −	1.2859e+2 (5.89e+1) −	1.9039e−1 (1.75e−1)
DTLZ2	3	12	1.9047e−1 (2.22e−2) −	1.5930e−1 (1.46e−4) −	1.1030e+0 (2.19e−1) −	1.3345e−1 (1.96e−3)
DTLZ3	3	12	6.9946e+1 (5.63e+1) −	1.8798e+1 (1.26e+1) −	8.3673e+2 (2.32e+2) −	1.2170e+1 (9.42e+0)
DTLZ4	3	12	2.8427e−1 (6.17e−2) +	4.5448e−1 (3.52e−1) +	1.3366e+0 (1.58e−1) −	7.0366e−1 (2.51e−1)
+/−/≈			1/3/0	1/3/0	0/4/0	
Average ranking			4.25	2.375	7	2

convergence or diversity and the other maintaining the final solution set.

In the experiment, μMOGAIF was compared with five state-of-the-art MOEAs and five micro population MOEAs on 28 benchmark problems, including DTLZ, ZDT, WFG, and MaF. The experiment results show that μMOGAIF is more competitive and effective in maintaining diversity than other MOEAs. In addition, the dual archive mechanism has the potential to deal with many-objective optimization problems. For the different proposed strategies, six algorithm variants verify their effectiveness of them. Finally, μMOGAIF is simulated on the low-power microprocessor.

Traditional MOEAs cannot be directly used on low-power microprocessors due to the limited resources, while its rich strategy can be reserved for population evolution. Based on a micro population scale, μMOGAIF is simulated on the low-power microprocessor and shows great performance on distribution. For future work, several avenues could be explored to improve μMOGAIF further, including: (1) Exploring Different Operator Selection Strategies: Investigating simpler yet more effective operator selection strategies could further enhance the adaptability and performance of the algorithm. (2) Refining the Dual Archive Mechanism: Additional research on the dual archive mechanism could optimize its effectiveness in balancing exploration and exploitation. (3) Exploring Different Population Evaluation Strategies: Researching more accurate and computationally simple population evaluation methods is more conducive to applying μMOEAs in microprocessors. These modifications could potentially enhance the algorithm efficiency, paving the way for more robust applications in resource-constrained environments like low-power microprocessors.

CRediT authorship contribution statement

Hu Peng: Conceptualization, Supervision, Funding acquisition. **Tian Fang:** Data curation, Methodology, Writing – original draft. **Jianpeng Xiong:** Methodology, Writing – review & editing. **Zhongtian Luo:** Data curation, Writing – review & editing. **Tao Liu:** Supervision, Funding acquisition. **Zelin Wang:** Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (62266024, 62367004).

Data availability

Data will be made available on request.

References

Abdi, Y., Asadpour, M., & Seyfari, Y. (2023). μMOSM: A hybrid multi-objective micro evolutionary algorithm. *Engineering Applications of Artificial Intelligence*, *126*, Article 107000.

Abdi, Y., & Feizi-Derakhshi, M.-R. (2020). Hybrid multi-objective evolutionary algorithm based on search manager framework for big data optimization problems. *Applied Soft Computing*, *87*, Article 105991.

Alba, E., & Dorronsoro, B. (2005). The exploration/exploitation tradeoff in dynamic cellular genetic algorithms. *IEEE Transactions on Evolutionary Computation*, *9*(2), 126–142.

Anshori, I., Mufiddin, G. F., Ramadhan, I. F., Ariasena, E., Harimurti, S., Yunkins, H., et al. (2022). Design of smartphone-controlled low-cost potentiostat for cyclic voltammetry analysis based on ESP32 microcontroller. *Sensing and Bio-Sensing Research*, *36*, Article 100490.

Aspray, W. (1997). The intel 4004 microprocessor: what constituted invention? *IEEE Annals of the History of Computing*, *19*(3), 4–15.

Babu, B., & Jehan, M. (2003). Differential evolution for multi-objective optimization. *Vol. 4*, In *2003 IEEE congress on evolutionary computation.* (pp. 2696–2703).

Behnamian, J., & Ghomi, S. F. (2014). Multi-objective fuzzy multiprocessor flowshop scheduling. *Applied Soft Computing*, *21*, 139–148.

Cabrera, J. C. F., & Coello, C. A. C. (2010). Micro-MOPSO: A multi-objective particle swarm optimizer that uses a very small population size. In *Multi-objective swarm intelligent systems: theory & experiences* (pp. 83–104).

Censor, Y. (1977). Pareto optimality in multiobjective problems. *Applied Mathematics Optimization*, *4*(1), 41–59.

Coello, C. A. C., & Pulido, G. T. (2001). A micro-genetic algorithm for multiobjective optimization. In *Evolutionary multi-criterion optimization* (pp. 126–140).

Deb, K., & Jain, H. (2014). An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, Part I: Solving problems with box constraints. *IEEE Transactions on Evolutionary Computation*, *18*(4), 577–601.

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, *6*(2), 182–197.

Deb, K., Thiele, L., Laumanns, M., & Zitzler, E. (2005). Scalable test problems for evolutionary multiobjective optimization. In *Evolutionary multiobjective optimization: theoretical advances and applications* (pp. 105–145).

Dutta, D., Sil, J., & Dutta, P. (2020). A bi-phased multi-objective genetic algorithm based classifier. *Expert Systems with Applications*, *146*, Article 113163.

Gary, S. (1996). Low-power microprocessor design. In *Low power design methodologies* (pp. 255–288). Boston, MA: Springer US.

Gasmi, I., Soui, M., Barhoumi, K., & Abed, M. (2024). Recommendation rules to personalize itineraries for tourists in an unfamiliar city. *Applied Soft Computing*, *150*, Article 111084.

Gee, S. B., Tan, K. C., & Abbass, H. A. (2017). A benchmark test suite for dynamic evolutionary multiobjective optimization. *IEEE Transactions on Cybernetics*, *47*(2), 461–472.

Goldberg, D. (1989). Sizing populations for serial and parallel genetic algorithms. In *Proceedings of the 3rd international conference on genetic algorithms* (pp. 70–79).

Hu, Y., Wang, J., Liang, J., Wang, Y., Ashraf, U., Yue, C., et al. (2022). A two-archive model based evolutionary algorithm for multimodal multi-objective optimization problems. *Applied Soft Computing*, *119*, Article 108606.

Huband, S., Hingston, P., Barone, L., & While, L. (2006). A review of multiobjective test problems and a scalable test problem toolkit. *IEEE Transactions on Evolutionary Computation*, *10*(5), 477–506.

Kashyap, S., Lakhani, C. S., Jain, D., & Agrawal, H. P. (2022). Intelligent security system using MSP430 microcontroller. In *2022 6th international conference on computing methodologies and communication* (pp. 659–663).

Kirimtat, A., Tasgetiren, M. F., Krejcar, O., Buyukdagli, O., & Maresova, P. (2024). A multi-objective optimization framework for functional arrangement in smart floating cities. *Expert Systems with Applications*, *237*, Article 121476.

Konak, A., Coit, D. W., & Smith, A. E. (2006). Multi-objective optimization using genetic algorithms: A tutorial. *Reliability Engineering & System Safety*, *91*(9), 992–1007.

Kukkonen, S., & Lampinen, J. (2005). GDE3: the third evolution step of generalized differential evolution. *Vol. 1*, In *2005 IEEE congress on evolutionary computation* (pp. 443–450).

Li, M., Yang, S., & Liu, X. (2014). Shift-based density estimation for Pareto-based algorithms in many-objective optimization. *IEEE Transactions on Evolutionary Computation*, *18*(3), 348–365.

Liu, G. P., Han, X., & Jiang, C. (2012). An efficient multi-objective optimization approach based on the micro genetic algorithm and its application. *International Journal of Mechanics & Materials in Design*, *8*(1), 37–49.

Lu, C., Gao, L., Li, X., Zeng, B., & Zhou, F. (2018). A hybrid multi-objective evolutionary algorithm with feedback mechanism. *Applied Intelligence: The International Journal of Artificial Intelligence, Neural Networks, and Complex Problem-Solving Technologies*, *48*, 4149–4173.

Mahto, K. K., Saha, S., & Das, P. (2019). Design of a solar MPP charge controller using mamdani FIS with generalized bell-shaped membership function. *International Journal of Recent Technology and Engineering (IJRTE)*, *8*(4), 8343–8348.

Maurya, V. K., & Nanda, S. J. (2023). Time-varying multi-objective smart home appliances scheduling using fuzzy adaptive dynamic SPEA2 algorithm. *Engineering Applications of Artificial Intelligence*, *121*, Article 105944.

Min, F., Hu, Q., & Zhu, W. (2014). Feature selection with test cost constraint. *International Journal of Approximate Reasoning*, *55*(1), 167–179.

Mokhtari-Moghadam, A., Salhi, A., Yang, X., Nguyen, T. T., & Pourhejazy, P. (2025). A multi-objective approach for the integrated planning of drone and robot assisted truck operations in last-mile delivery. *Expert Systems with Applications*, *269*, Article 126434.

Nag, K., Pal, T., & Pal, N. R. (2015). ASMiGA: An archive-based steady-state micro genetic algorithm. *IEEE Transactions on Cybernetics*, *45*(1), 40–52.

Nebro, A. J., Luna, F., Alba, E., Dorronsoro, B., Durillo, J. J., & Beham, A. (2008). AbYSS: Adapting scatter search to multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, *12*(4), 439–457.

Peng, H., Kong, F., & Zhang, Q. (2024). Micro multiobjective evolutionary algorithm with piecewise strategy for embedded-processor-based industrial optimization. *IEEE Transactions on Cybernetics*, *54*(8), 4763–4774.

Peng, H., Li, L., Mei, C., Deng, C., Yue, X., & Wu, Z. (2023). Gene expression programming with dual strategies and neighborhood search for symbolic regression problems. *Applied Soft Computing*, *145*, Article 110616.

Peng, H., Wang, C., Han, Y., Xiao, W., Zhou, X., & Wu, Z. (2022). Micro multi-strategy multi-objective artificial bee colony algorithm for microgrid energy optimization. *Future Generation Computer Systems*, *131*, 59–74.

Peng, H., Xiao, W., Han, Y., Jiang, A., Xu, Z., Li, M., et al. (2022). Multi-strategy firefly algorithm with selective ensemble for complex engineering optimization problems. *Applied Soft Computing*, *120*, Article 108634.

Peng, H., Xiong, J., Pi, C., Zhou, X., & Wu, Z. (2024). A dynamic multi-objective optimization evolutionary algorithm with adaptive boosting. *Swarm and Evolutionary Computation*, *89*, Article 101621.

Pulido, G., & Coello, C. (2003). The micro genetic algorithm 2: Towards online adaptation in evolutionary multiobjective optimization. (pp. 252–266).

Quinn, H., Fairbanks, T., Tripp, J. L., Duran, G., & Lopez, B. (2014). Single-event effects in low-cost, low-power microprocessors. In *2014 IEEE radiation effects data workshop* (pp. 1–9).

Santiago, A., Dorronsoro, B., Fraire, H. J., & Ruiz, P. (2020). Micro-genetic algorithm with fuzzy selection of operators for multi-objective optimization: μFAME. *Swarm and Evolutionary Computation*, *61*(1), Article 100818.

Sindhya, K., Miettinen, K., & Deb, K. (2013). A hybrid framework for evolutionary multi-objective optimization. *IEEE Transactions on Evolutionary Computation*, *17*(4), 495–511.

Storn, R., & Price, K. (1997). Differential evolution – A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, *11*(4), 341–359.

Tian, Y., Cheng, R., Zhang, X., & Jin, Y. (2017). PlatEMO: A MATLAB platform for evolutionary multi-objective optimization. *IEEE Computational Intelligence Magazine*, *12*(4), 73–87.

Tiwari, S., Fadel, G., & Deb, K. (2011). AMGA2: improving the performance of the archive-based micro-genetic algorithm for multi-objective optimization. *Engineering Optimization*, *43*(4), 377–401.

Tiwari, S., Koch, P., Fadel, G., & Deb, K. (2008). AMGA: an archive-based micro genetic algorithm for multi-objective optimization. In *Conference on genetic & evolutionary computation* (pp. 729–736).

Wang, H., Jiao, L., & Yao, X. (2015). Two_Arch2: An improved two-archive algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Computation*, *19*(4), 524–541.

Wang, D., & Ma, L. (2024). Ranking-based multi/many-objective evolutionary optimization with hierarchical evaluation rules and its application in water distribution network. *Applied Soft Computing*, *162*, Article 111807.

Yang, S., Li, M., Liu, X., & Zheng, J. (2013). A grid-based evolutionary algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Computation*, *17*(5), 721–736.

Yu, G., Ma, L., Jin, Y., Du, W., Liu, Q., & Zhang, H. (2022). A survey on knee-oriented multi-objective evolutionary optimization. *IEEE Transactions on Evolutionary Computation*, *26*(6), 1452–1472.

Zhang, Q., & Li, H. (2007). MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, *11*(6), 712–731.

Zhang, Y., Li, B., Hong, W., & Zhou, A. (2023). MOCPSO: A multi-objective cooperative particle swarm optimization algorithm with dual search strategies. *Neurocomputing*, *562*, Article 126892.

Zhou, A., Qu, B. Y., Li, H., Zhao, S.-Z., Suganthan, P. N., & Zhang, Q. (2011). Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation*, *1*(1), 32–49.

Zitzler, E., Deb, K., & Thiele, L. (2000). Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation*, *8*(2), 173–195.

Zitzler, E., & Künzli, S. (2004). *Indicator-Based Selection in Multiobjective Search* (pp. 832–842).

Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C. M., & Da Fonseca, V. G. (2003). Performance assessment of multiobjective optimizers: an analysis and review. *IEEE Transactions on Evolutionary Computation*, *7*(2), 117–132.