GMHA: Growable Meta-Heuristic Algorithm for Multi-Objective Optimization Problems and its Application in Cloud Scheduling

Guangyao Zhou, Minxian Xu *Senior Member, IEEE*, Jiashu Zhang*, Rajkumar Buyya *Fellow, IEEE*,

***Abstract*—Scheduling problems in distributed computing (such as Cloud or edge computing) usually belong to multi-objective optimization problems (MOPs). Meta-heuristic algorithm (MHA) is an effective type of contemporary algorithm for solving difficult MOPs. The improvement of MHA is an urgent spot, whose main challenge lies in accelerating convergence speed while enhancing the optimality of convergence solutions. This challenge requires a universal and effective method to improve the search efficiency of various MHAs throughout the entire iteration process for solving MOPs. Towards this target, this paper restructures the MHAs' framework uniformly, and proposes a universal growable meta-heuristic algorithm framework (GMHA) with hybrid multi-growth routes. Providing the flexibility for the combination of various algorithms to serve as the growth route, GMHA is applicable to diverse MHAs. For the sake of the adaptability of GMHA for various MOPs including Cloud scheduling, the paper further establishes several general growth routes, including equidistant feasible solution search route (EFSS), gradient neighborhood search route (GNS), and weighted neighborhood search route (WNS). Statistical test on various MOP benchmarks demonstrate that GMHA has a probability of 90.15% to improve the performance of various MHAs. Compared to the corresponding MHA in experiments on multi-objective Cloud scheduling problems, GMHA achieves 2.116 times the average convergence speed, with the specific reductions of 25.05% in total energy consumption, 4.94% in the maximum energy consumption of server nodes, and 35.12% in the sum of the standard deviations of utilizations.**

***Index Terms*—Growable Meta-Heuristic, Multi-Objective Optimization Problem, Cloud Scheduling, Energy Consumption**

I. Introduction

DISTRIBUTED computing systems (e.g., Cloud computing) that provide flexible services have always been confronted with multi-objective optimization problems (MOPs) [1]. An MOP typically contains multiple decision variables, leading to a huge search space and challenging optimization methods, which emerges as a widespread attraction followed by scientific technology and theory [2], [3].

In MOP, a feasible algorithm is required to continuously optimize the non-dominated solution set to approach the true Pareto front (PF). Therefore, the method of searching for a single solution in the neighborhood is generally not suitable for MOPs. Meta-heuristic algorithm (MHA) has two epitome characteristics: establishing the fitness evaluation system based on multi-objectives to compare and screen multiple solutions; and regenerating the next iteration's solution based on specific strategies [4]. Therefore, MHAs can achieve the search for multiple optimized solutions and also become promising methods for solving complex MOPs [5], which have also been widely applied to Cloud scheduling [6]–[8].

However, meta-heuristic algorithms still face significant challenges: 1) premature convergence (local optimum) with a single search process [7]; 2) insufficient search results in poor optimality of solutions, with few optimal Pareto solutions [4]; 3) consuming computing power on some redundant solutions [9]. The ultimate manifestation is the poor average solving efficiency and large time complexity, specifically for large-scale Cloud scheduling. Especially, the constantly increasing decision-making space highlights these challenges.

At present, there are hundreds of MHAs [9]. However, most existing improvement methods or strategies are only designed for one or a few types of MHAs. It is an urgent challenge to find a method with universality for a wide range of MHAs and MOPs, so as to enhance the overall capability of MHAs.

With the above considerations, this paper aims to explore a universal strategy that can widely improve the performance of various MHAs holistically. Through collation and reorganization of the existing MHAs' frameworks, we have observed that almost all MHAs have two identical or similar processes: generating new solutions (offspring solutions) based on a certain meta-heuristic strategy (called meta-heuristic searching operation), and completing the current iteration to re-generate the initial solutions of the next iteration (called update operation). The specific modes of update operation and searching operation may vary with the type of MHAs, but both processes are generally indispensable. This phenomenon enlightens us to consider whether it is possible to enhance the performance of various MHAs by adding a universal strategy between these two processes. Therefore, this paper restructures the MHAs' framework uniformly, adds a growth stage after the previous iteration's update operation and before the current meta-heuristic searching operation, and then proposes a universal growable meta-heuristic algorithm framework (GMHA). GHMA allows some solutions to be improved through additional strategies before participating in meta-heuristic searching, significantly improving the efficiency of the algorithm. GMHA is applicable to various MHAs, as long as they contain these two processes (meta-heuristic searching operation and update operation). In experiments

Guangyao Zhou (guangyao_zhou@swjtu.edu.cn) and Jiashu Zhang (jszhang@home.swjtu.edu.cn) are with School of Computing and Artificial Intelligence, Southwest Jiaotong University, China.

Minxian Xu (mx.xu@siat.ac.cn) is with Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China.

Rajkumar Buyya (rbuyya@unimelb.edu.au) is with Cloud Computing and Distributed Systems (CLOUDS) Laboratory, Department of Computing and Information Systems, the University of Melbourne, Australia.

of statistical tests using various GMHAs on various MOP benchmarks, GMHAs can quickly find a better solution set, consuming fewer iterations compared to the state-of-the-art MHAs. Moreover, experiments in Cloud scheduling demonstrate its practicality.

The main contributions of this paper are as follows:

(1) **Growable meta-heuristic algorithm framework (GMHA):** this paper extracts the common processes of various MHAs and reconstructs a universal framework of them. With the universal MHA framework, we add a growth stage before the meta-heuristic search process and propose GMHA. GHMA allows the designated solutions to be improved by additional strategies before participating in meta-heuristic searching, significantly improving the efficiency of the entire algorithm. GMHA is a universal framework that adapts to various MHAs.

(2) **Hybrid multi-growth routes:** to solve various MOPs, especially in Cloud scheduling, this paper proposes several general growth routes, including equidistant feasible solution search route (EFSS), gradient neighborhood search route (GNS), and weighted neighborhood search route (WNS). Ulteriorly, this paper proposes the hybrid multi-growth routes strategy integrating various growth routes, which possesses the improvement effect in various MOPs.

(3) **Various instantiated growable meta-heuristic algorithms:** introducing state-of-the-art meta-heuristic algorithms into GMHA framework, this paper instantiates multiple specific growable meta-heuristic algorithms, expanding the capability boundary of MHAs.

(4) Extensive experiments on various MOPs benchmarks and Cloud scheduling scenarios verify the adaptability of GMHA to various MHAs and MOPs, as well as demonstrate the superiority and practicability of the proposed algorithms to Cloud scheduling.

The remainder of this paper is organized as follows: we review the related work in Section II; the methodologies with the GMHA framework, general growth routes and various instantiated GMHAs are proposed in Section III; we present the experiments to study the performance of GMHA framework in Section IV; finally, we conclude this paper in Section V.

II. Related Work and Motivation

This section briefly reviews the related work from two aspects: meta-heuristic algorithms and MOPs in Cloud computing. Then, this section explains the motivation of this paper through discussions and analysis of related work.

A. Meta-heuristic Algorithms

A meta-heuristic algorithm (MHA) is generally considered as an algorithm with certain adaptability to solving problems, which iteratively searches solutions by mimicking intelligent processes or behaviors [4]. The types of MHAs vary with the strategies of search, forms of solution sets and the manner of the iteration process. The frequently applied MHAs for MOPs are population-based algorithms. Shown as Fig. 1, population-based algorithms mainly include four categories: evolutionary algorithms, swarm intelligence algorithms, physics-based algorithms, human-based algorithms, etc [10], [11].

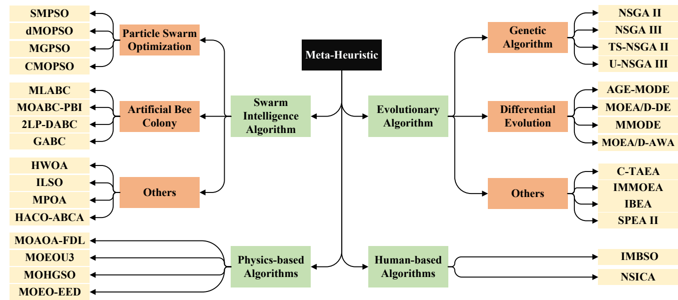


Fig. 1. The categories of meta-heuristic algorithms for MOPs.

The evolutionary algorithm generates new solutions by selection, mutation, and crossover operators to evolve existing solutions [9], [12]. Evolutionary algorithms have spawned various contemporary algorithms for MOPs, including NSGA (non-dominated sorting genetic algorithm), MODE (multi-objective differential evolution), etc [13]–[15]. Some typical NSGAs include NSGA II, NSGA III, U-NSGA III (unified NSGA III) [13], [16], etc. Some examples of MODE include AGE-MODE [2], MOEA/D-DE [15], MOEA/D-HSS [3], etc. Other evolutionary algorithms for MOPs include CTAEA (two-archive evolutionary algorithm for constrained MOP) [17], IBEA (indicator-based evolutionary algorithm) [18], SPEA II (strength Pareto evolutionary algorithm) [19], etc.

A swarm intelligence algorithm usually contains multiple intelligent agents (called a swarm) with certain self-organizing abilities, and searches for a solution set based on the collective behavior or status of the swarm [9], [11], [20]. Some prevalent swarm intelligence algorithms are particle swarm optimization (PSO), artificial bee colony algorithm (ABC), ant colony optimization (ACO), etc [20], [21]. PSO-based multi-objective algorithms include SMPSO (speed-constrained multi-objective PSO) [22], MOPSO_MCD (multi-objective PSO with modified crowding distance) [23], etc. Some ABC-based algorithms are MOABC-PBI [24], GABC (grid-based ABC) [25], etc. Other swarm intelligence algorithms for MOPs include HACO-ABCA (hybrid ACO and ABC algorithm) [26], HWOA (hybrid whale optimization algorithm) [27], etc.

Other categories of MHAs can also be leveraged for MOPs, but with relatively low frequencies in existing research [9]. Generally, physics-based and human-based algorithms require the assistance of other strategies to solve MOPs. Some examples include MOEOU3 (multi-objective optimization algorithm consisting of equilibrium optimizer and NSGA-III) [28], MOEO-EED (multi-objective equilibrium optimizer with exploration-exploitation dominance strategy) [29], PETMSA (Pareto entropy-based two-mode multi-objective simulated annealing algorithm) [30], NSICA (multi-objective non-dominated imperialist competitive algorithm) [31], etc.

B. MHA-based MOPs in Cloud Scheduling

The factors that directly affect the difficulty of solving optimization problems include the numbers and forms of objectives, decision variables, and constraint conditions. In addition, some important factors are the properties of the search space and the properties of the theoretical optimal solution set (discrepancy degree, concavity, monotonicity), not directly reflected in the problems' expression, but closely related to the difficulty of solving the problem.

For MOPs in Cloud-related distributed systems, the community is paying significant attention. Lin et al proposed Peak Efficiency Aware Scheduling (PEAS) to optimize the energy consumption and QoS in the online virtual machine (VM) allocation of Cloud [32]. Zhou et al used evolutionary multitasking optimization based on a meta-heuristic idea to schedule constrained workflows in Cloud [8]. Shen et al applied Cuckoo search to solve the task scheduling and cache updating [33]. Monge et al designed an online auto-scaler based on the genetic algorithm to optimize VMs workflows in Cloud systems [34]. Bilal et al designed a distributed grey wolf optimizer to schedule workflows in Cloud environments [35]. Shikha et al designed two variants of whale optimization algorithms for efficiently placing VMs in Cloud computing [36]. Sun et al proposed a new genetic programming with multi-tree representation to automatically learn the task, Cloud and resource selection rules simultaneously for the dynamic flexible workflow scheduling in multi-Clouds [37]. Yang et al combined machine learning and greedy algorithms to schedule workflow tasks within unknown task execution time [38]. Xiao et al constructed an efficient service-aware VMs scheduling approach based on the MOEA to simultaneously optimize communication cost and migration time [39]. Qiu et al applied an MPGA (multi-population-based genetic algorithms) to solve multi-objective workflow scheduling to optimize makespan and energy consumption simultaneously [40]. Kaleibar et al established a customized genetic algorithm to optimize SLA-aware service provisioning in Cloud [6].

C. Motivation

From related work, solving MOPs mainly relies on population-based MHAs, especially on evolutionary algorithms and swarm intelligence algorithms. The communities mainly concentrate on the strategies for decomposition of problems, evaluation metrics, screen method, and regeneration mechanism. However, these improvement measures are generally effective for only one or a few algorithms, lacking universality to various MHAs. Furthermore, MHAs have been widely leveraged to solve MOPs in Cloud scheduling, where MOPs exhibit variable complexity with the composition of their specific elements in different scenarios. Therefore, the existing improvement measures for MHAs have unstable adaptability to a wide range of Cloud scheduling.

The above circumstances prompt this paper to explore a universal method to improve various MHAs in different MOPs. A universal strategy is necessary and significant to improve the overall performance of MHAs in solving Cloud scheduling. Based on this motivation, this paper sorts out the architectures and processes of various MHAs, extracts their commonalities, and reconstructs a universal framework of MHAs. With the reconstructed framework, this paper establishes a universal growable meta-heuristic algorithm framework (GMHA) with hybrid multi-growth routes, which can pervasively improve the performance of various MHAs in various MOPs, demonstrating adaptability to multi-objective Cloud scheduling.

III. Methodologies

Assuming there are n decision variables $\vec{X} = (x_1, x_2, \ldots, x_n)$, m objectives $\vec{F}(\vec{X})$ and l constraint conditions $\vec{S}(\vec{X}) = (s_1, s_2, \ldots, s_l)$, MOP can be defined as:

$$\begin{cases} \text{Minimize } \vec{F}(\vec{X}) = \left(\omega_1(\vec{X}), \omega_2(\vec{X}), \ldots, \omega_m(\vec{X})\right), \\ \text{subject to} : s_j(\vec{X}) \leq 0, j = 1, 2, \ldots, l. \end{cases} \tag{1}$$

When $m > 1$, a single optimization solution is usually unable to satisfy the minimization of all optimization objectives. Thus, a MOP requires obtaining multiple solutions that will form a non-dominated solution set (denoted as D) according to the dominant relationship between solutions. Eq. (1) can represent most problems in Cloud scheduling. For two solutions X and Y, if $\omega_i(\vec{X}) \leq \omega_i(\vec{Y})$ for $\forall i \in \{1, 2, \ldots, m\}$, and $\exists k \in \{1, 2, \ldots, m\}$ **s.t.** $\omega_k(\vec{X}) < \omega_k(\vec{Y})$, it is defined that $\vec{X}$ dominates $\vec{Y}$. A non-dominated solution set obeys that $\vec{X}$ and $\vec{Y}$ do not dominate each other for $\forall \vec{X} \neq \vec{Y} \in D$.

To explore a universal method for improving performance in MOPs, we also consider a multi-objective Cloud scheduling problem $\min \omega = \min \left(\omega^{(1)}, \omega^{(2)}, \omega^{(3)}\right)$, in addition to the general MOP forms of Eq. (1). The objectives are respectively as minimizing total energy consumption (denoted as $\min \omega^{(1)}$, optimizing energy consumption), minimizing the maximum energy consumption of each server node (denoted as $\min \omega^{(2)}$, balancing energy consumption) and minimizing the sum of the standard deviations of multi-dimensional resource utilization (denoted as $\min \omega^{(3)}$, load balancing). The formula for energy consumption introduces nonlinear terms as follows:

$$\begin{aligned} E_j = &\sum_{k=1}^{l} \left(a_{jk} \left(\sum_{i=1}^{n} x_{ij} u_{ijk} \right)^2 + b_{jk} \left(\sum_{i=1}^{n} x_{ij} u_{ijk} \right) \right) \\ &+ \sum_{k=1}^{l} \left(c_{jk} + d_{jk} \max_{i=1}^{n} (x_{ij}) \right) \end{aligned} \tag{2}$$

where E_j means the energy consumption of the j-th server node; l is number of resource dimensions; n is the number of tasks; u_{ijk} means the resource occupancy rate of the i-th task for the k-th dimension in j-th server node; a_{jk}, b_{jk}, c_{jk} and d_{jk} are the coefficients of energy consumption for quadratic polynomials; the matrix $\{x_{ij}\}_{1 \leq i \leq n, 1 \leq j \leq m}$ consists of decision variables, where $x_{ij} \in \{0, 1\}$ and m is the number of server nodes. $x_{ij} = 1$ means the i-th tasks is allocated to the j-th server node. Then, multiple objectives can be written as Eq. (3) where $\operatorname*{std}_{j=1}^{m}$ is standard deviations.

$$\begin{cases} \min \omega^{(1)} = \min \sum_{j=1}^{m} E_j, \quad \min \omega^{(2)} = \min \max_{j=1}^{m} E_j, \\ \min \omega^{(3)} = \min \sum_{k=1}^{l} \operatorname*{std}_{j=1}^{m} \left(\sum_{i=1}^{n} x_{ij} u_{ijk} \right). \end{cases} \tag{3}$$

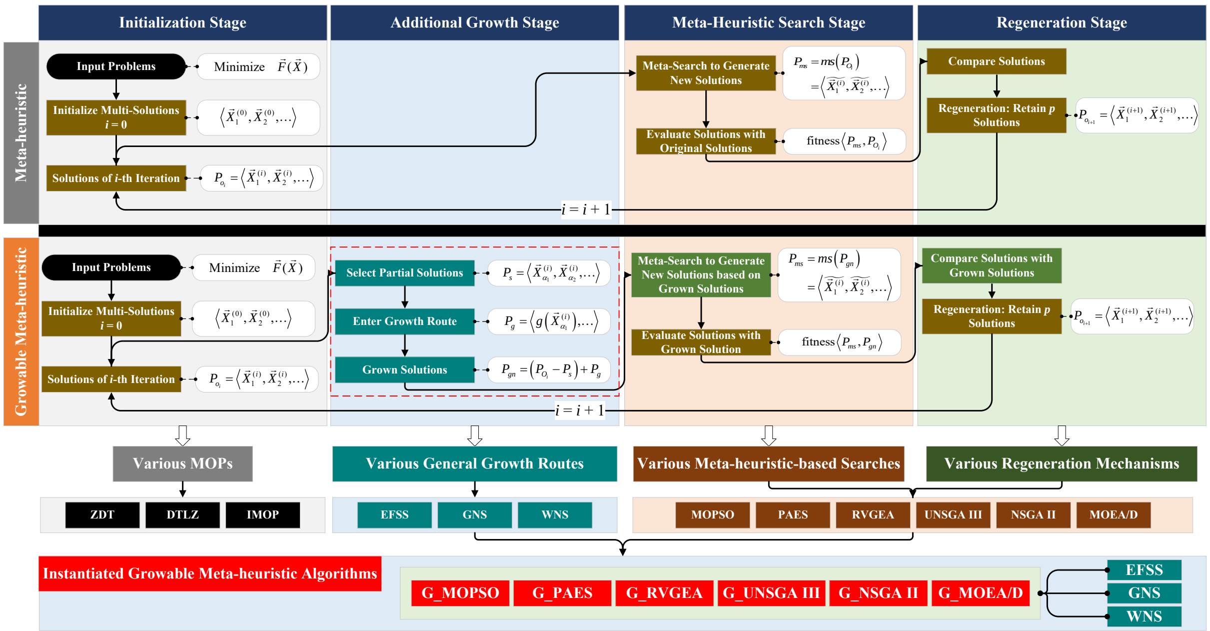


Fig. 2. The universal frameworks of growable meta-heuristic algorithms compared with universal framework of meta-heuristic algorithms.

To achieve the improvement on the performance of algorithms for MOP problems, especially for Cloud scheduling, our proposed methodologies mainly consist of four components: a reconstructed framework of MHAs, a universal framework of GMHA with hybrid multi-growth routes, multiple general growth routes, and instantiated algorithms of GHMA.

The frameworks of methodologies are shown in Fig. 2. The meta-heuristic row of Fig. 2 is the reconstructed framework of MHAs containing three stages: initialization stage, meta-heuristic search stage and regeneration stage. On this basis, we add an additional growth stage, thus forming a growable meta-heuristic algorithm framework (GMHA) as shown in Fig. 2. To adapt to different MOPs, this paper further proposes several general growth routes, including equidistant feasible solution search route (EFSS), gradient neighborhood search route (GNS), and weighted neighborhood search route (WNS). Incorporating different MHAs based on the GMHA framework, this paper instantiates multiple growable meta-heuristic algorithms, including growable multi-objective particle swarm algorithm (G_MOPSO), growable Pareto archived evolution strategy (G_PAES), growable constrained two archive evolutionary algorithm (G_CTAEA), growable non-dominated sorting genetic algorithm II (G_NSGA II), etc.

Next, we respectively present the four components in detail and conduct theoretical analysis on the proposed GMHA.

A. Universal Meta-heuristic Algorithm Framework

The universal framework and strategy are indispensable to improve the overall performance of various meta-heuristic algorithms (MHAs) in solving different MOPs, especially in Cloud scheduling. Therefore, this paper investigates the existing MHAs and reconstructs them by universally dividing them into three stages: initialization stage, meta-heuristic search stage and regeneration stage. The reconstructed framework of

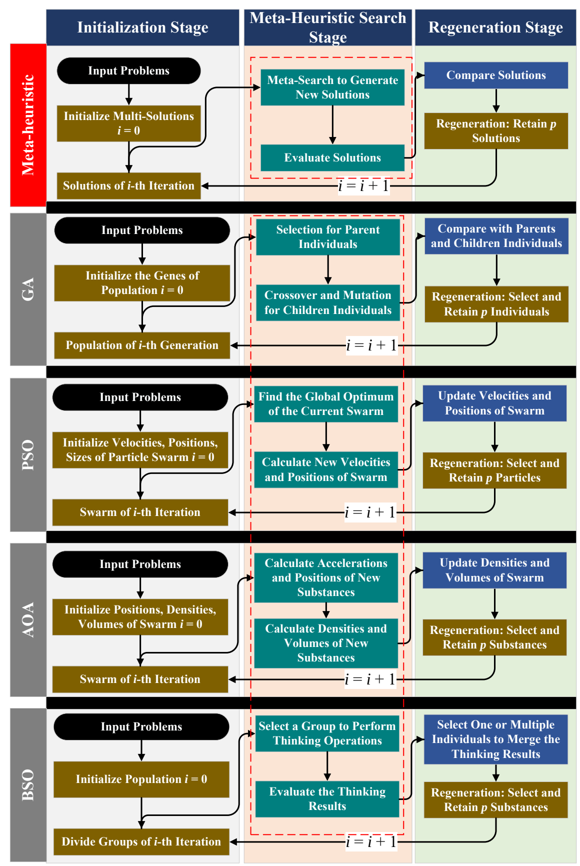


Fig. 3. The flowcharts of the universal meta-heuristic framework with the mapping relationships diagram to several popular meta-heuristic algorithms (PSO, GA, GSO, BSO).

MHAs is shown in the first row of Fig. 3. In order to present the reconstruction process of the universal framework more intuitively, we select several popular MHAs, including PSO (particle swarm optimization, belonging to swarm intelligence algorithm), GA (genetic algorithm of evolutionary algorithm), AOA (Archimedes optimization algorithm of physics-based algorithm) and BSO (brain storm optimization of human-based algorithm), as well as draw their flowcharts and mapping relationships with the universal meta-heuristic framework.

From Fig. 3, one of the most important processes of MHAs, is the meta-heuristic search process based on the specific strategies, which also determines the search capability. E.g., in Fig. 3, the search process of GA is based on crossover and mutation, and PSO is based on the position update of the particle swarm. In addition to the algorithms in Fig. 3, other MHAs can also make similar mappings with the universal framework. These search processes can be collectively referred to as meta-heuristic search processes (called the meta-heuristic search stage). Moreover, all iteration-based MHAs require updating the current active solution (or solution set) at the end of each iteration, which can be called the regeneration stage.

B. Growable Meta-heuristic Algorithm Framework

Comprehensive enhancement to the performance of MHAs for MOPs requires an architecture level improvement. To achieve it, this paper considers adding an additional growth stage on top of the three stages (initialization stage, meta-heuristic search stage and regeneration stage) of the universal MHA framework. Then, we propose the growable meta-heuristic algorithm framework, shown in Fig. 2.

Different from original MHAs, GMHA selects a part of solutions to enter the growth routes for improvement in the growth stage, and then substitutes the grown solutions into the subsequent meta-heuristic search stage. GMHA framework has considerable flexibility, allowing for the combination of different growth routes and MHAs. To ensure the universality of GMHA for optimization problems, GMHA applies hybrid growth routes instead of the single growth route. Due to varying optimization degrees of the active solution set, the improvement effect of the growth route also differs. Benefit from the flexibility of framework, GMHA allows flexible configuration of growth strategies for each iteration and solution. Thus, it is not necessary to use the same growth strategy at every iteration, and each solution also does not have to use the same growth routes. The flowchart of hybrid multi-growth route strategy in GMHA is shown in Fig. 4.

As shown in Fig. 4, the hybrid growth strategy of GMHA selects a subset from the initial solution set of the current iteration to participate in growth; makes the selected solutions enter different growth routes where the growth routes assigned to different individuals can change with iteration; integrates solutions after growth and solutions without growth to form the mature solutions; and substitutes mature solutions into the meta-heuristic search stage sequentially.

For the sake of organizing the execution process and the subsequent theoretical analysis, we present the pseudo-code of GMHA with hybrid multi-growth routes in Algorithm 1. What is more flexible in Algorithm 1 is that each iteration's meta-heuristic strategy can be different, for example, the first iteration uses the genetic algorithm's search strategy, while the second iteration can still use the particle swarm optimization algorithm's search strategy. In addition, the growth routes can also be dynamically adjusted based on the context during the optimization process, such as leveraging game theory and policy-based reinforcement learning. However, this paper mainly focuses on the construction of a growable meta-heuristic algorithm framework rather than the heterogeneous meta-heuristic search strategies and growth routes. Therefore, this paper only considers the case where each iteration of the meta-heuristic search strategy is fixed with randomly selecting a growth route from the given general growth routes.

Algorithm 1: GMHA with hybrid multi-growth routes

Input : Objectives $\vec{F}(\vec{X})$ and constraints $\vec{S}(\vec{X})$
Output: Non-dominated solution set D

Initialization Stage:
- Initialize solutions $P_{O_0} = \left\langle \vec{X}_1^{(0)}, \ldots, \vec{X}_p^{(0)} \right\rangle$ where p is a preset swarm (or population) size, and set of growth route functions as $Gr = \langle g_1, g_2, \ldots \rangle$.
- Record the non-dominated solution set as D.

for $i = 0$ *to iteration number* **do**
- **Growth Stage with Hybrid Growth:**
  - Select partial solutions from P_{O_i} as $P_S = \left\langle \vec{X}_{\alpha_1}^{(i)}, \vec{X}_{\alpha_2}^{(i)}, \ldots, \vec{X}_{\alpha_{Q_i}}^{(i)} \right\rangle$, to randomly enter different growth route functions in Gr where the vector composed of the growth route functions assigned to each solution is denoted as $Ga = \langle g_{\beta_1}, g_{\beta_2}, \ldots, g_{\beta_{Q_i}} \rangle$ where $g_{\beta_k} \in Gr$ for $1 \leq k \leq Q_i$;
  - Then obtain the solutions after growth as $P_g = \left\langle g_{\beta_1}\left(\vec{X}_{\alpha_1}^{(i)}\right), g_{\beta_2}\left(\vec{X}_{\alpha_2}^{(i)}\right), \ldots, g_{\beta_{Q_i}}\left(\vec{X}_{\alpha_{Q_i}}^{(i)}\right) \right\rangle$.
  - Integrate the solutions after growth (P_g) and the solutions without growth ($P_{O_i} - P_s$) to obtain mature solutions as $P_{gn} = (P_{O_i} - P_s) + P_g$.
- **Meta-heuristic Search Stage:**
  - **Solving Offspring Solutions:**
    - Based on the mature solutions P_{gn}, use a specific meta-heuristic search strategy (or multiple strategies) to obtain the offspring solutions as $P_{ms} = ms\left(P_{gn}\right)$.
  - Stack the offspring solutions and the mature solutions as $stack\left\langle P_{ms}, P_{gn} \right\rangle$.
  - Calculate the fitness of the stacked solutions as $fitness\left\langle P_{ms}, P_{gn} \right\rangle$ according to dominance relationship (domination hierarchy, crowding distance, etc) based on the objectives $\vec{F}(\vec{X})$.
- **Regeneration Stage:**
  - Substitute the stacked solutions to update the non-dominated solution set D.
  - Compare the stacked solutions and retain p solutions ($P_{O_{i+1}}$) to the next iteration.

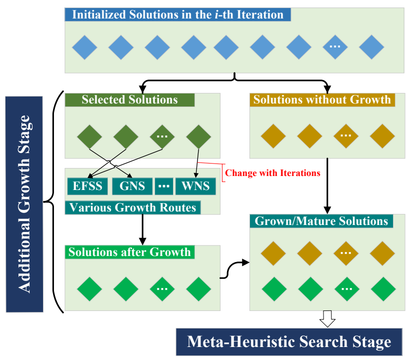


Fig. 4. The flowchart of hybrid multi-growth route strategy in GMHA.

C. General Growth Routes for MOPs

From the above presentation, an indispensable component of GMHA is the growth route. MHAs have an important advantage, that is, various MHAs generally have certain adaptability to different MOPs. In order to maintain this advantage for GHMA, we need to construct some growth routes that are universal for various MOPs. In this paper, we design three general growth routes, i.e., equidistant feasible solution search route (EFSS), gradient neighborhood search route (GNS), and weighted neighborhood search route (WNS).

(1) Equidistant feasible solution search route (EFSS): assuming a solution to growth is $\vec{X} = (x_1, x_2, \ldots, x_n)$, EFSS randomly selects a decision variable x_l, sets values equidistant within its domain, and obtain multiple values of x_i as $x_l = \{e_1, e_2, \ldots, e_u\}$ where e_1 is the lower bound, e_u is the upper bound of x_l. The multiple values needs to satisfy $e_{j+1} - e_j = \Delta x$ for $\forall j \in \{1, 2, \ldots, u-1\}$ where the interval Δx can be customized. There are u neighbor solutions according to the different values of x_l. EFSS selects the solutions with the highest fitness from these u solutions as the growth solution.

(2) Gradient neighborhood search route (GNS): GNS randomly selects one (or more) decision variable x_l from $\vec{X} = (x_1, x_2, \ldots, x_n)$, starts from the value of x_l to add or subtract the interval Δx, and obtains multiple values $\{\ldots, x_l - 2\Delta x, x_l - \Delta x, x_l, x_l + \Delta x, x_i + 2\Delta x, \ldots\}$ to replace x_l in $\vec{X}$. Then, GNS also selects the solutions with the highest fitness as the growth solution.

(3) Weighted neighborhood search route (WNS): WNS randomly generates an n-dimensional unit weight $\vec{W} = \langle w_1, w_2, \ldots, w_n \rangle$ for a solution, and obtain multiple solutions as $\left\{\vec{X}, \vec{X} + \Delta x \vec{W}, \vec{X} + 2\Delta x \vec{W}, \ldots\right\}$. Then, WNS also selects the solutions with the highest fitness.

The advantage of these three growth routes is that they are applicable to various MOPs. As they only consider the range of values for decision variables and they are not affected by the optimization objectives, which is sufficient to support the validation of GMHA's universality. These three routes can also be further improved, such as selecting multiple decision variables at once for similar operations, which can also keep the universality to various MOPs. Since the target of this paper is to propose GMHA and verify its universality for various types of MHAs and MOPs, this paper only considers the local growth of one decision variable per growth step.

D. Instantiations of GMHA

With GMHA framework and general growth routes, another specific component is the method of meta-heuristic search. Owing to the flexibility of the GMHA framework shown as Fig. 2 and Algorithm 1, switching the meta-heuristic search process to a specific MHA (such as GA, PSO, BSO, etc.) can obtain a corresponding growable meta-heuristic algorithm. To instantiate the GMHA in this paper, we select several typical MHAs, including the multi-objective particle swarm algorithm (MOPSO), Pareto archived evolution strategy (PAES), constrained two archive evolutionary algorithm (CTAEA), reference vector guided evolutionary algorithm (RVGEA), unified non-dominated sorting genetic algorithm III (UNSGA III), etc., and built their corresponding growable meta-heuristic algorithm. Due to the similarity in the position of various meta-heuristic algorithms in GMHA framework (namely the meta-heuristic search part of Algorithm 1), this paper only presents the pseudo-code of G_MOPSO in Algorithm 2. Other meta-heuristic algorithms that did not appear in the experiments of this paper, such as Distributed Grey Wolf Optimizer (DGWO) [41], island-based Cuckoo search with polynomial mutation (iCSPM) [42], etc., can also be used to obtain the corresponding growable algorithms in a similar way, where the main difference lies in the method of solving offspring solutions, i.e., meta-heuristic search strategy.

Algorithm 2: Growable multi-objective particle swarm optimization algorithm (G_MOPSO)

```
Input  : Objectives F(X) and constraints S(X)
Output: Non-dominated solution set D
Initialization Stage:
    Similar to initialization stage in Algorithm 1.
for i = 0 to iteration number do
    Growth Stage with Hybrid Growth:
        Similar to growth stage in Algorithm 1.
    Meta-heuristic Search Stage:
        Solving Offspring Solutions:
            Find the global optimum solutions in the
              mature solutions P_gn.
            Calculate the new velocities and positions
              of the mature solutions P_gn, and obtain
              the offspring solutions according to new
              velocities and positions.
        Stack the offspring solutions and the mature
          solutions, and calculate their fitness.
    Regeneration Stage:
        Similar to regeneration stage in Algorithm 1.
```

E. Theoretical Analysis of GMHA

To further illustrate the advantages of GMHA in theory, we analyze its complexity and convergence speed.

The total complexity (denoted as c) of an instantiated algorithm of GMHA can be written as Eq. (4), which mainly contains the complexity of meta-heuristic search in each iteration (denoted as c_{ms}), the complexity of the additional growth route in each iteration (denoted as c_g), and the total number of iterations (denoted as κ).

$$c = \kappa \left(c_{ms} + c_g\right) \tag{4}$$

To further analyze the complexity of GMHA, we can discuss decisive components (c_{ms} and c_g) of Eq. (4).

(1) The complexity of meta-heuristic search in each iteration (c_{ms}) is determined by the number of active individuals (i.e., p), the complexity of calculating fitness of each individual (denoted as c_f) and the complexity of updating populations (denoted as c_u). Among them, c_f is determined by the relationship between the optimization objective function and the solution set, as well as the fitness indicators. When the problem and fitness indicators (such as dominance level, crowding degree, etc.) are given, c_f can be regarded as a constant. c_u is determined by meta-heuristic strategies and competitive elimination mechanisms in population updates. When these strategies are given, c_u can also be considered as a constant. Therefore, the complexity of meta-heuristic search in each iteration can be expressed as Eq. (5).

$$c_{ms} = \mathbf{O}\left(p \cdot \left(c_f + c_u\right)\right) \tag{5}$$

(2) The complexity of growth stage in each iteration (c_g) is determined by the growth quota of each iteration (denoted as Q_i which means the number of solutions entering additional growth routes in the i-th iteration), the growth step of each iteration (denoted as S_i which means the times of each solution undergoing searches along the growth route), and the complexity of a solution to grow once through growth route (denoted as c_r). c_r is related to optimization problems and the algorithm of the growth route, which can also be regarded as a constant for the given MOP and growth route. Then, the complexity of the growth stage in each iteration can be expressed as Eq. (6).

$$c_g\left(Q_i, S_i\right) = \mathbf{O}\left(S_i \cdot Q_i \cdot c_r\right) \tag{6}$$

Substituting Eq. (5) and Eq. (6) into Eq. (4) can obtain the detailed complexity of GMHA as Eq. (7).

$$c = \mathbf{O}\left(\kappa \cdot p \cdot \left(c_f + c_u\right) + c_r \cdot \sum_{i=1}^{\kappa}\left(S_i \cdot Q_i\right)\right) \tag{7}$$

For different basic meta-heuristic search strategies, additional growth routes, and specific forms of MOPs, the coefficients (c_f, c_u and c_r) in Eq. (7) will have specific expressions. E.g., for growable NSGA II using GNS (with fixed parameters $S_i = S$ and $Q_i = Q$) as the growth route to solve the problem $\omega^{(3)}$, the complexity is $\mathbf{O}\left(\kappa\left(p^3 + p \cdot m \cdot n \cdot l + Q \cdot S \cdot m \cdot n \log n\right)\right)$. For the general form of c in Eq. (7), when the growth parameters S and Q are far smaller than p, the complexity of GMHA approximately equals that of MHA as $c \approx c_{ms}$. However, the additional growth route does introduce more computational complexity, so further discussion is needed on the optimization degree under unit complexity to analyze the improvement effect of GMHA on the convergence speed. As this paper aims to establish a relatively universal analysis method, we will discuss it in the general form of Eq. (4) or Eq. (7).

Subsequently, we start with analyzing the optimization benefits of a certain iteration. The initial solution set in the i-th iteration can be denoted as $P_{O_i} = \left\langle \vec{X}_1^{(i)}, \vec{X}_2^{(i)}, \ldots, \vec{X}_p^{(i)} \right\rangle$, the function evaluating the optimization degree is denoted as ζ. c_{ms} remains relatively constant in each iteration. Thus, for MHA shown as the meta-heuristic row in Fig. 2, the improvement in the optimization degree (denoted as D_{ms}) of the solution set after the meta-heuristic search process in the i-the iteration can be expressed as:

$$D_{ms} = \zeta\left(ms\left(P_{O_i}\right)\right) - \zeta\left(P_{O_i}\right). \tag{8}$$

where $ms\left(P_{O_i}\right)$ means the solution set obtained from solution set P_{O_i} through a once meta-heuristic search. Therefore, the improvement in optimization degree obtained by unit computing complexity (denoted as ϱ_{ms}) can be expressed as

$$\varrho_{ms} = \frac{D_{ms}}{c_{ms}} = \frac{\zeta\left(ms\left(P_{O_i}\right)\right) - \zeta\left(P_{O_i}\right)}{p \cdot \left(c_f + c_u\right)}. \tag{9}$$

Regardless of the expression or encoding in any meta-heuristic, the individual can accurately transform into the form of the solution for the targeted optimization problem. This property indicates that individuals of different MHAs can be transformed into each other or can be represented uniformly. Due to the fact that the searchability of MHAs is affected by the quality of the current active solution set, if some solutions can be deterministically improved through a certain strategy (with less computational power consumption), the overall search efficiency of MHAs can be enhanced. Therefore, on the basis of the universal framework of MHAs, adding a growth route and selecting partial solutions for additional improvement, a universal growable meta-heuristic algorithm framework (GMHA) can be obtained as the growable meta-heuristic row in Fig. 2. In GMHA, solutions selected to participate in growth are denoted as $P_s = \left\langle \vec{X}_{\alpha_1}^{(i)}, \vec{X}_{\alpha_2}^{(i)}, \ldots, \vec{X}_{\alpha_{Q_i}}^{(i)} \right\rangle$. Thus, the solution set after growth from P_s can be represented as $P_g = \left\langle g\left(\vec{X}_{\alpha_1}^{(i)}\right), g\left(\vec{X}_{\alpha_1}^{(i)}\right), \ldots, g\left(\vec{X}_{\alpha_{Q_i}}^{(i)}\right) \right\rangle$. It can be obtained that after the additional growth stage, the solution set changes from P_{O_i} to $P_{gn} = \left(P_{O_i} - P_s\right) + P_g$. Therefore, the improvement in the optimization degree (denoted as $D_{gms}\left(Q_i, S_i\right)$) of the solution set after the growable meta-heuristic search process in the i-th iteration is:

$$\begin{aligned} D_{gms}\left(Q_i, S_i\right) &= \zeta\left(ms\left(P_{gn}\right)\right) - \zeta\left(P_{O_i}\right) \\ &= \zeta\left(ms\left(\left(P_{O_i} - P_s\right) + P_g\right)\right) - \zeta\left(P_{O_i}\right) \end{aligned}. \tag{10}$$

The improvement in optimization degree obtained by unit computing complexity for GHMA (ϱ_{gms}) can be expressed as:

$$\begin{aligned} \varrho_{gms} &= \frac{D_{gms}\left(Q_i, S_i\right)}{c_{ms} + c_g\left(Q_i, S_i\right)} \\ &= \frac{\zeta\left(ms\left(\left(P_{O_i} - P_s\right) + P_g\right)\right) - \zeta\left(P_{O_i}\right)}{p \cdot \left(c_f + c_u\right) + S_i \cdot Q_i \cdot c_r}. \end{aligned} \tag{11}$$

To make GMHA superior to MHA, there only needs to be a combination of Q_i and S_i such that $\varrho_{gms} > \varrho_{ms}$, which mainly depends on the growth route and meta-heuristic strategy. Through theoretical deduction, we can obtain Theorem 1, which can be leveraged for quick verification or selection of growth routes.

Theorem 1: For solutions (or solution sets) that have not reached convergence, when the growth route satisfies Eq. (12), then $\exists\,(S_i, Q_i)$ **s.t.** GMHA superior to MHA.

$$\frac{D_{gms}\left(Q_i,1\right)-D_{gms}\left(Q_i,0\right)}{c_g\left(Q_i,1\right)} > \frac{D_{gms}\left(Q_i,0\right)}{c_{ms}} \tag{12}$$

Theorem 1 can be proved through the method of limits.

Proof 1: When $S_i = 0$, GMHA will degrade into MHA. Thus, $\lim_{S_i\to 0} D_{gms}\left(Q_i,S_i\right) = D_{ms}$. Adding the numerator and denominator on both sides of Eq. (12) separately must result a fraction between the two, so Eq. (13) is tenable.

$$\begin{aligned}\frac{D_{gms}\left(Q_i,1\right)-D_{gms}\left(Q_i,0\right)}{c_g\left(Q_i,1\right)} &> \frac{D_{gms}\left(Q_i,1\right)}{c_g\left(Q_i,1\right)+c_{ms}} \\ &> \frac{D_{gms}\left(Q_i,0\right)}{c_{ms}}\end{aligned} \tag{13}$$

Therefore,

$$\varrho_{gms}\left(Q_i,1\right) > \varrho_{gms}\left(Q_i,0\right) = \varrho_{ms}. \tag{14}$$

When $S_i \to +\infty$, the algorithm of the growth route will achieve convergence. Therefore, $\exists Y \in \mathbb{N}$ **s.t.**: for $\forall y \geq Y$, $D_{gms}\left(Q_i,y\right) = D_{gms}\left(Q_i,Y\right)$. Thus, we can obtain Eq. (15).

$$\lim_{y\to+\infty} \varrho_{gms}\left(Q_i,y\right) = 0 < \varrho_{ms} = \varrho_{gms}\left(Q_i,0\right) \tag{15}$$

Combining Eq. (14) and Eq. (15) can obtain that: $\exists Y \in \mathbb{N}^+$ **s.t.** $\varrho_{gms}\left(Q_i,Y\right) \geq \varrho_{gms}\left(Q_i,y\right)$ is tenable for $\forall y \in \mathbb{N}$. In this case, there must be an optimal parameter configuration for S_i and Q_i of GMHA, which maximizes the absolute improvement in optimization degree per unit of computing complexity.

Theorem 1 explains one of the conditions for the superiority of GMHA, which is actually easy to achieve under the flexible configuration of growth routes. Therefore, it also demonstrates the universal superiority of GMHA. The principle of GMHA lies in sufficiently combining the global search-ability of meta-heuristics and the directional search-ability (in the gradient direction or directional derivative) of growth routes, so as to improve the overall convergence speed and optimization degree of the algorithm. Moreover, the additional growth process requires little computing power to improve a small part of the solutions, which can enhance the overall performance of the entire meta-heuristic search process. The above theoretical analysis is not limited to a specific meta-heuristic strategy, growth route, or optimization problem, so it can be extended to various scenarios, which demonstrates the scalability of our proposed GMHA. In the next section, we will also verify the universality of GMHA through experiments.

IV. Experimental Studies

A. Experiment Setting

For the sake of the comprehensive evaluations of our proposed GMHA framework, we carry out several groups of experiments from the following aspects:

(1) universality tests for various meta-heuristic algorithms and various MOP benchmarks;
(2) practicability tests for multi-objective Cloud scheduling;
(3) statistical tests via the Wilcoxon test and Friedman test.

In order to verify the universal improvement effect of GHMA on different MHAs, we select representative MHAs as the basic algorithms, and construct their growable meta-heuristic algorithms leveraging the framework in Fig. 2 and Algorithm 1. The basic MHAs in the experiment include 12 algorithms, i.e., SPEA II, PAES, UNSGA III, NSGA II, RVGEA, CTAEA, NSGA III, MOEAD, IBEA_FC, MOPSO, SMPSO, and GREA (Grid-based Evolutionary Algorithm) [16], [17], [19], [43]. To avoid early convergence and sufficiently reflect the search-ability of algorithms, we combine domination hierarchy and crowding distance [13] to evaluate the fitness of solutions during the meta-heuristic search stage.

These algorithms and their growable algorithms are evaluated on multiple recognized MOPs testing benchmarks, including ZDT [44], WFG [45], DTLZ [46] and IMOP [47]. This combination allows for the inclusion of diverse forms of Pareto front in the testing benchmarks (covering convex, concave, mixed, disconnected, 2D and 3D PFs). The representative indicators adopted to evaluate the performance of the algorithms are IGD (inverted generational distance) and HV (hyper-volume) [48]. IGD can be used to measure the distance between the Pareto front of the non-dominated solution set and the true (theoretical) Pareto front. The formulas of IGD and HV can be written as:

$$\text{IGD} = \frac{\sum_{i=1}^{|PF|} d_i^2}{|PF|}, \quad \text{HV} = \delta \bigcup\nolimits_{i=1}^{|S|} v_i \tag{16}$$

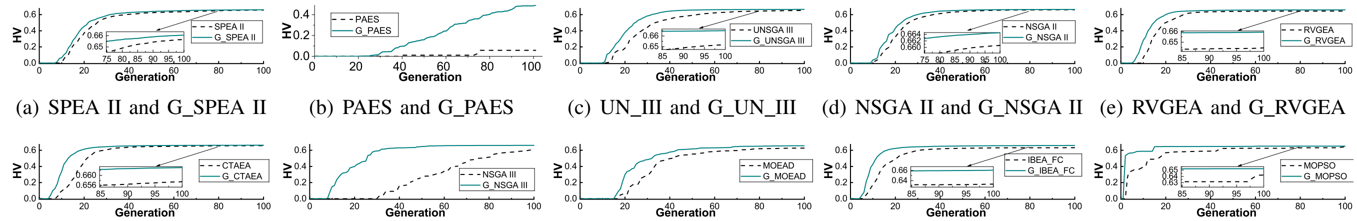
where $|PF|$ is the number of true Pareto front, d_i represents the Euclidean distance between the i-th reference objective vector and nearest objective values obtained by the algorithm, δ represents Lebesgue measure, $|S|$ means the number of non-dominated solutions, v_i represents the hyper-volume formed by the reference point and the i-th solution in the non-dominated solution set.

To maintain the comparability of the results, we use the absolute HV and absolute IGD of non-dominated solutions related to the true Pareto front by calling the functions of **pymoo.indicators.igd.IGD** and **pymoo.indicators.hv.Hypervolume** [48], whose reference points are set as the corresponding points of the true Pareto front. The growth quota and growth step parameters of GMHA are respectively set as $Q_i = 10$ and $S_i = 2$. Other parameter settings, such as population size and iteration times, may vary in each experimental group and will be explained in the corresponding experimental group section.

B. Universality Tests for Basic Algorithms and Problems

1) Universality for Meta-heuristic Algorithms: The experiments in this subsection are to verify the prompting effect of GHMA on the basis of different meta-heuristic algorithms from the perspective of hyper-volume over iterations.

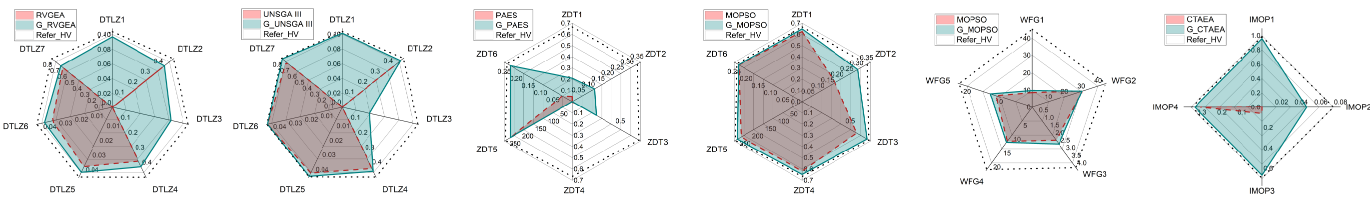
To ensure that the experimental results are conducted under consistent parameter configurations, we set the population size



(a) SPEA II and G_SPEA II (b) PAES and G_PAES (c) UN_III and G_UN_III (d) NSGA II and G_NSGA II (e) RVGEA and G_RVGEA

(f) CTAEA and G_CTAEA (g) N_III and G_N_III (h) MOEAD and G_MOEAD (i) IB_FC and G_IB_FC (j) MOPSO and G_MOPSO

Fig. 5. Verification of GMHA's universality for various meta-heuristic algorithms on test benchmark ZDT1 (100 population size, 100 generations).



(a) RVGEA and G_RVGEA in DTLZ (b) UNSGA III and G_UNSGA III in DTLZ (c) PAES and G_PAES in ZDT (d) MOPSO and G_MOPSO in ZDT (e) MOPSO and G_MOPSO in WFG (f) NSGA II and G_NSGA II in WFG

Fig. 6. Radar charts of HV to verify GMHA's universality for various test benchmarks (100 population size, 100 generations).

of each algorithm to 100, and observe the trend of their hyper-volume within 100 iterations. Due to similar comparison conclusions under different benchmarks, only a subset of the experimental results is presented in this paper, where we selected the experimental results under the ZDT1 problem. The experimental results of each basic MHA and its corresponding GMHA are shown in Fig. 5.

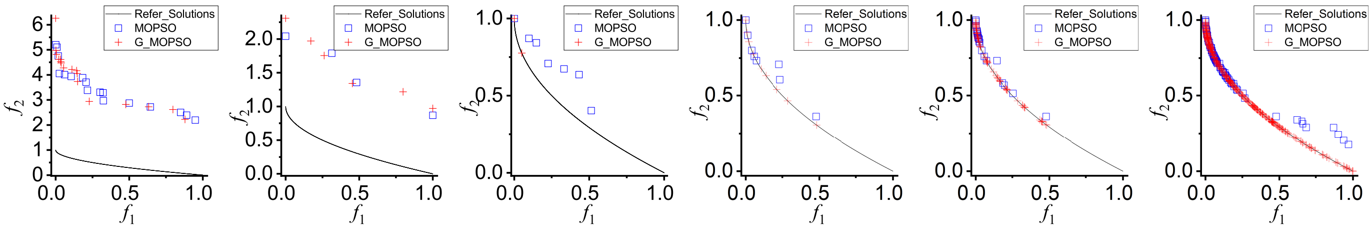
In the comparison results of each sub-figure in Fig. 5, the HV-over-iteration curve of the GMHA is higher than that of its basic MHA. The results of Fig. 5 show that within the same number of iterations, the non-dominated solution set of MOPs obtained by a GMHA is closer to the true solution set compared to its basic MHA. This phenomenon indicates that different MHAs have improved their convergence speed and optimization performance in multi-objective optimization to a certain extent after introducing the proposed GMHA framework, which demonstrates the universal implications of GMHA for multiple MHAs. In addition, due to the different solving processes and strategies of different basic MHAs, their basic performance varies. On the basis of the performance of different basic MHAs, the improvement effect presented by the GMHA framework is also different. For example, in Fig. 5, when utilizing PAES to solve the ZDT1 problem, the HV-over-iteration curve is the lowest. After leveraging GMHA, the performance has been significantly improved. However, due to the limitations of the basic performance of PAES, the performance of G_PAES with the addition of GMHA still has a significant gap compared to other algorithms. It indicates that the upper improvement effect of the GMHA is limited by the performance of the basic MHA. However, it also indirectly reflects the robustness of GMHA's improvement effect, that even if the performance of the basic algorithm is too low or too high, GMHA can still present a certain degree of improvement effect, showing the significance of GMHA in enhancing the overall capability of MHAs.

GHMA possessing universality for various MHAs may be because GMHA framework is a further improvement on top of the universal MHA framework (Fig. 3) that is reconstructed by extracting the common process of various MHAs. Therefore, the improvement effect of GMHA actually reflects the improvement of GMHA on the universal framework, and it is precisely the universality of the universal MHA framework that generalizes the improvement effect of GMHA.

2) Universality for Optimization Problems: The above experimental results verify the universality of GMHA for different basic meta-heuristic algorithms only in ZDT1. The experiments in this subsection are to verify the universality of GHMA for different MOPs.

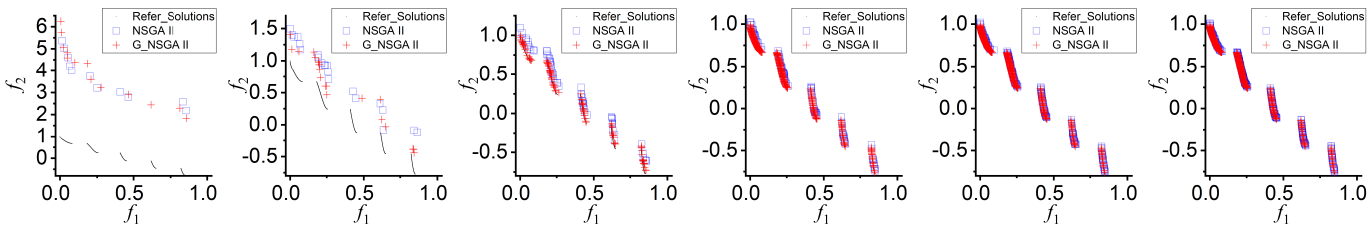
To verify this property, we select several sets of MHAs and their growable meta-heuristic algorithms for experiments under different benchmarks, and record their HV at the 100-th generation. As different parameter settings can derive similar conclusions, we only present experimental results for the configuration of 100 population size. Then, we plot the experimental results under the same benchmark into radar charts, as shown in Fig. 6, where "Refer_HV" means the HV of the true Pareto front.

Fig. 6 presents experimental results of RVGEA, UNSGA III, PAES, MOPSO, NSGA II and their GMHAs under multiple benchmarks. From the figures, the HV has been generally improved under various MOP benchmarks by GMHA. It is worth noting that due to the different patterns of the solution sets in different MOPs, the performance of the same algorithm varies greatly for different benchmarks. On the basis of different basic performances, GMHA has a wide range of improvement effects. Especially in some difficult problems, the performance of the original basic algorithm is poor, while the indicators of the growable algorithms can approach the theoretical optimal HV (reference HV). For example, RVGEA in DTLZ3, UNSGA III in DTLZ1, and MOPSO in ZDT2 can



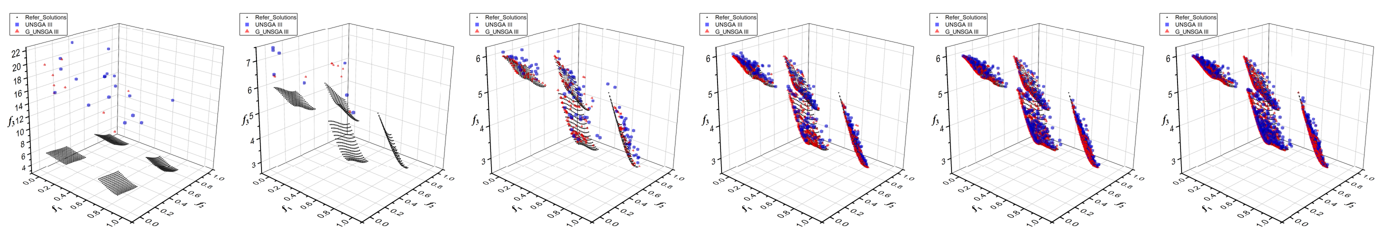
(a) 0-th iteration (b) 2-nd iteration (c) 4-th iteration (d) 6-th iteration (e) 8-th iteration (f) 10-th iteration

Fig. 7. Non-dominated solutions over iterations of MOPSO and G_MOPSO in ZDT1 (400 population size).



(a) 0-th iteration (b) 20-th iteration (c) 40-th iteration (d) 60-th iteration (e) 80-th iteration (f) 100-th iteration

Fig. 8. Non-dominated solutions over iterations of NSGA II and G_NSGA II in ZDT3 (100 population size).



(a) 0-th iteration (b) 20-th iteration (c) 40-th iteration (d) 60-th iteration (e) 80-th iteration (f) 100-th iteration

Fig. 9. Non-dominated solutions over iterations of UNSGA III and G_UNSGA III in DTLZ7 (112 population size: 6 reference size).

only obtain a solution set with HV significantly worse than the theoretical optimal HV, while their corresponding GMHA successfully achieve appreciable HV close to the theoretical optimal HV.

GHMA possessing universality for various MOPs may mainly come from the universality of the proposed general growth routes expounded in Section III-C. The general growth routes are direct optimization processes within a specific local range of the selected variables, ensuring to some extent that the evaluation indicators of the solution obtained after passing through the general growth routes are at least not inferior to those of the solution before growth. Moreover, these direct optimization processes are applicable to most representative optimization problems, finally proving that GMHA has universality for various MOPs.

However, the proposed GMHA also has limitations in improving certain problems, such as WFG1. It indicates that there is still room for improvement in certain specific problems, such as designing targeted growth routes or dynamically configuring growth strategies for specific problems. It is also because this paper adopts growth routes with certain universality in the GMHA framework. Therefore, the improvement for different problems may not be consistent. The adaptability of general routes to different problems varies, also indicating that the performance of GMHA is limited by the usage of general growth routes, which can be regarded as sacrificing superiority over a specific single problem in exchange for universality over extensive problems. Overall, GMHA still demonstrates considerable universality and effective improvement.

3) Discussion on Convergence via Non-dominated Solutions: The above experiment verified the universality of the proposed GMHA for meta-heuristic algorithms and optimization problems, where the main indicator is HV. HV-over-iteration can, to some extent, demonstrate the convergence performance of algorithms. However, it is also necessary to further observe from the perspective of the non-dominated solution set-based Pareto scatter, which can intuitively exhibit the changes in non-dominated solution sets. As a large number of experimental results can derive similar conclusions, we only present a subset of them. We present the results for three representative benchmark problems leveraging different algorithms: MOPSO and G_MOPSO for ZDT1 (2D, convex, continuous), NSGA II and G_NSGA II for ZDT3 (2D, discrete), UNSGA III and G_UNSGA III for DTLZ7 (3D, discrete). Then, the non-dominated solutions over iterations are respectively plotted in Fig. 7 to Fig. 9.

As G_MOPSO in ZDT1 converges in a very small number of iterations, Fig. 7 only provides results within 10 iterations by setting 400 population size. According to the results of the 2-nd iteration in Fig. 7(b), the difference between MOPSO's and G_MOPSO's non-dominated solution sets is not obvious, and moreover, their non-dominated solution sets are both far from the true Pareto front. At the 4-th iteration, both MOPSO's and G_MOPSO's non-dominated solution sets achieve close to the true Pareto front, but the overall number of non-dominated solutions is relatively small and only one or two solutions fall on the true Pareto front. As iteration increases in Fig. 7(e) to Fig. 7(f), G_MOPSO shows a better trend of more non-dominated solutions falling on the Pareto front than MOPSO. Until the 10-th iteration, G_MOPSO obtains a non-dominated solution set almost covering the true Pareto front, while MOPSO can only cover the upper left corner (coverage ratio is less than 0.5). Furthermore, starting from the 4-th iteration, the probability of G_MOPSO's non-dominated solutions falling on the Pareto front reaches 100% and all non-dominated solutions remain on the Pareto front throughout the subsequent iterations. However, MOPSO always has a portion of non-dominated solutions that are obviously far from the Pareto front. The change process of results from Fig. 7(a) to Fig. 7(f) clearly demonstrates that G_MOPSO has better convergence and can quickly obtain more solutions on the true Pareto front. This improvement mainly comes from the GMHA framework with additional growth routes.

Due to more iterations required for each algorithm to reach convergence under the corresponding problem in the experiments of Fig. 8 and Fig. 9, we present their flowcharts of the changes in non-dominated solutions within 100 iterations (100 population size or a value close to 100). Results in Fig. 8 and Fig. 9 also clearly demonstrate the comparative relationship between MHAs and their corresponding GMHAs. For example, G_NGSA II's non-dominated solutions almost cover the true Pareto front at the 40-th iteration of Fig. 8(c), while NGSA II only has few solutions falling on the Pareto front; G_UNSGA III covers most of the PF at the 60-th iteration of Fig. 8(d), while UNSGA III's non-dominated solutions are mostly not on PF.

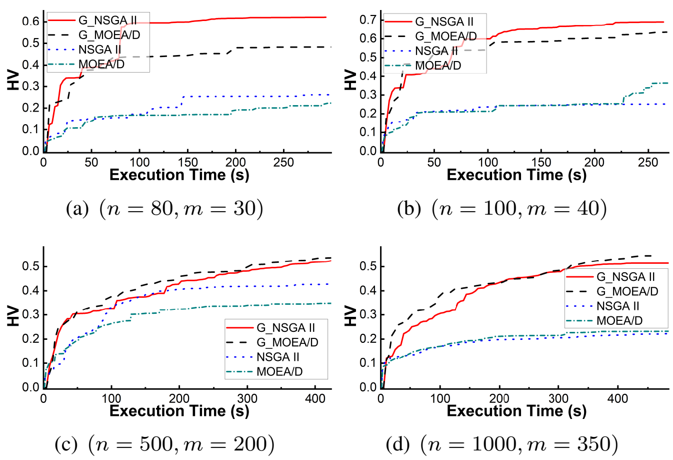
From the perspective of Pareto front patterns, it can be seen that the adaptability of GMHA to MOPs is actually reflected in its adaptability to changes in different PF patterns including 2D to 3D, convex to non-convex, continuous to discrete.

C. Application for Cloud Scheduling

To verify the practicability of our proposed GMHA framework for multi-objective problems in Cloud scheduling, we carry out experiments in the multi-objective problem $\min \omega = \min\left(\omega^{(1)}, \omega^{(2)}, \omega^{(3)}\right)$. The experiments take the resource utilization rates of three dimensions and their relationship with energy consumption into account. The coefficients of energy consumption are generated as integers according to the uniform distributions:

$$\begin{cases} a_{jk} \sim \mathrm{U}(1,10), \quad b_{jk} \sim \mathrm{U}(0,100), \\ c_{jk} \sim \mathrm{U}(100,200), \quad d_{jk} \sim \mathrm{U}(500,1000). \end{cases} \tag{17}$$

Considering NSGA and MOEA/D are representative enough to reflect the properties of population-based meta-heuristics, we observe the performance of G_NSGA II and G_MOEA/D by comparing to their basic algorithms NSGA II and MOEA/D. Among them, the growable meta-heuristic algorithms leverage the proposed hybrid growth routes to additionally adjust the solutions, where the growth quota is 5 and the growth step is 5. To accelerate the overall execution of the optimization algorithm, we run the calculation function for individual fitness on the GPU (RTX 2080 Ti), achieving parallel evaluation for multiple solutions. To ensure the comparison under the same computing power consumption, we set the evolution generations of G_NSGA II and G_MOEA/D to 100, as well as adopt the results with the corresponding execution time. To unify the scale of the three optimization objectives so as to reflect the influence of each optimization objective in HV, we implement normalization by calling the function **pymoo.indicators.hv.Hypervolume** [48] in settings of "zero_to_one = True" and "norm_ref_point = True".



(a) $(n = 80, m = 30)$ (b) $(n = 100, m = 40)$ (c) $(n = 500, m = 200)$ (d) $(n = 1000, m = 350)$

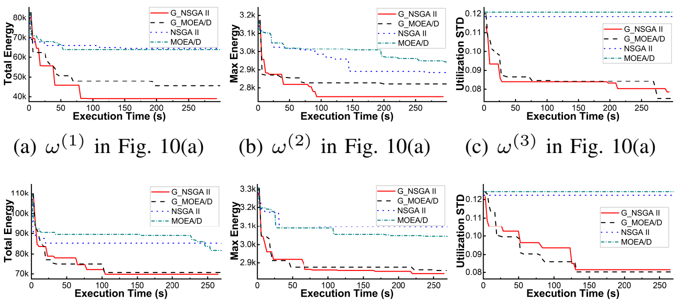
Fig. 10. HV over execution time for problem Eq. (3) in Cloud scheduling comparing G_NSGA II and G_MOEA/D with NSGA II and MOEA/D.

As multiple groups of experiments will lead to similar conclusions, we only present a subset of results, including $(n = 80, m = 30)$, $(n = 100, m = 40)$, $(n = 500, m = 200)$ and $(n = 1000, m = 300)$. Then, we can plot the HV over the execution time of algorithms in Fig. 10. The obvious observation in Fig. 10 is that the HV curve of the GMHAs is significantly higher than that of the corresponding basic algorithms, which demonstrates the adaptability of our proposed GMHA framework to application scenarios and optimization problems. Especially for large-scale optimization problems in Cloud computing (with hundreds or thousands of decision variables), introducing a small amount of additional search during the optimization solution process (consuming only a bit of computing power) can fully retain the global search-ability of the meta-heuristic algorithm while adding the trend of convergence, enabling the algorithm to obtain better solutions in less time. Taking the ratio between the HV of GMHA and the corresponding MHA at the same execution time in Fig. 10 to represent the ratio of convergence speed (abbreviated as RCS), and taking the ratio between the HV of the convergent solution to represent the ratio of the optimization degree (abbreviated as ROD), we can list the results in Table I. From

Table I, GMHA has 2.116 times the average convergence speed and 2.068 times the optimization degree of MHA in the Cloud scheduling scenarios.

TABLE I
RATIOS OF CONVERGENCE SPEED AND OPTIMIZATION DEGREE BETWEEN GMHA AND THE CORRESPONDING MHA.

Ratio	GMHA	Fig. 10(a)	Fig. 10(b)	Fig. 10(c)	Fig. 10(d)	Average
RCS	G_NSGA II	2.623	2.509	1.197	2.102	2.108
	G_MOEA/D	2.560	2.350	1.442	2.144	2.124
ROD	G_NSGA II	2.372	2.741	1.236	2.314	2.166
	G_MOEA/D	2.243	1.746	1.543	2.350	1.971



(a) $\omega^{(1)}$ in Fig. 10(a) (b) $\omega^{(2)}$ in Fig. 10(a) (c) $\omega^{(3)}$ in Fig. 10(a)

(d) $\omega^{(1)}$ in Fig. 10(b) (e) $\omega^{(2)}$ in Fig. 10(b) (f) $\omega^{(3)}$ in Fig. 10(b)

Fig. 11. Optimization results of each objective over execution time for scenarios of $(n = 80, m = 30)$ and $(n = 100, m = 40)$.

For the sake of observation of each optimization objective, we plot the corresponding time-history curves of scenarios of $(n = 80, m = 30)$ and $(n = 100, m = 40)$ in Fig. 11. From Fig. 11, our proposed GMHA not only achieves a better optimization degree of multi-objectives, but also ensures the fast search speed of each single-objective. Compared to the corresponding MHA, GMHA achieves improvement in each objective, averagely reducing the total energy consumption ($\omega^{(1)}$) by 25.05%, the maximum energy consumption of each node ($\omega^{(2)}$) by 4.94%, and the sum of the standard deviations of multi-dimensional resource utilization ($\omega^{(3)}$) by 35.12%.

The above experiments verify the practicability of our proposed GMHA for large-scale Cloud scheduling considering multiple objectives, such as energy consumption and utilization. The above experiments, as a sample of resource scheduling, are sufficiently representative of Cloud computing, which aims at providing flexible services to various users.

D. Discussion on Statistical Tests

The above experiments mainly analyze and discuss the results of one instance. In order to further comprehensively evaluate the improvement effect of GHMA framework on various MHAs in various MOPs, this subsection analyzes and discusses the statistical test of multiple instances of experimental results. In statistical tests, all algorithms are run 30 times independently for each test benchmark. For the sake of enriching the diversity expression of indicators, this subsection mainly presents the results under the IGD indicator (where the HV indicator has similar qualitative comparative conclusions).

We utilize the Wilcoxon rank sum test and Friedman test to evaluate the algorithms, whose results are respectively presented in Table II and Table III. The Wilcoxon rank-sum test can test the null hypothesis that two sets of measurements are drawn from the same distribution. At the 5% test level, two algorithms have obvious differences in a function if the p-value $\leq$ 0.05, otherwise the difference is not obvious. On the basis of the Wilcoxon rank-sum test, we carry out a comprehensive Wilcoxon rank-sum test to evaluate all 24 algorithms respectively. In the comprehensive Wilcoxon rank-sum test, the results of each algorithm come from taking the current algorithm as the target algorithm and subjecting it to the 1-on-1 Wilcoxon rank-sum test with the other 23 algorithms respectively. Based on the results of the 1-on-1 Wilcoxon rank-sum test, it is decided to add 1 to the corresponding W/T/L term. For example, the 15/1/7 in column ZDT4 of row SPEA II in Table II indicates that SPEA II has won 15 algorithms, is close to 1 algorithm, and has lost to 7 algorithms.

From Table II and Table III, the statistical test results obtained by GMHAs are generally better than those of their corresponding basic meta-heuristic algorithms, which verifies the superiority of GMHA framework from the perspective of statistical test, reflecting the stability of the GMHA in improving the performance of basic meta-heuristic algorithms.

However, there are still some specific problems where the comprehensive Wilcoxon test results of some MHAs and their corresponding GMHAs are close or even the same. One reason may be that we only select the IGD values of the 50-th iteration for comparison, and some basic MHAs and their corresponding GMHAs may have reached similar convergence states (approaching theoretical states) in the 50-th iteration with close IGD values. This reason mainly corresponds to the cases where the comprehensive Wilcoxon test results of the basic algorithm and its growable algorithm are very considerable, such as (SPEA II, G_SPEA II) in IMOP4, and (MOPSO, G_MOPSO) in ZDT4. The other reason may be that not all three general growth routes proposed in this paper are suitable for the specific scenario (specific algorithms in specific problems). The strategy adopted in this paper is to randomly select one route from the three general growth routes for growth in each iteration, so the final effect of GMHA may be that the increase and decrease effects cancel each other out. This reason corresponds to the cases where the basic algorithm and its growable algorithm both perform poorly, such as (MOPSO, G_MOPSO) in ZDT2, and (SMPSO, G_SMPSO) in DTLZ2. This phenomenon indirectly reflects the disadvantage of the GMHA based on the hybrid multi-growth route proposed in this paper, lacking targeted consideration for flexible configuration of growth routes in specific scenarios, which is also an issue deserving further studies in future work.

According to Table III, the Friedman test can also indicate that GMHA has a stable improvement effect on MHA. Generally, in the Friedman test, GMHA has a 90.15% (i.e., $\frac{238}{264}$) probability of improving the ranking of MHA. The statistical test results can be used to conduct further analysis that the introduction of hybrid multi-growth routes results in excellent overall improvement. Relying solely on a single route will result in lower adaptability coverage than hybrid multi-growth routes. Moreover, due to the universality of the GMHA being

TABLE II
COMPREHENSIVE WILCOXON RANK-SUM TEST (W/T/L) BASED ON IGD AT THE 50-TH ITERATION FOR 30 TIMES RUNS (100 POPULATION SIZE).

Algorithm	ZDT						WFG					DTLZ							IMOP			
	1	2	3	4	5	6	1	2	3	4	5	1	2	3	4	5	6	7	1	2	3	4
SPEA II	17/3/3	18/0/5	19/1/3	15/1/7	20/1/2	11/8/4	17/3/3	18/3/2	14/6/3	17/2/4	17/3/3	10/2/11	19/2/2	11/2/10	21/0/2	19/3/1	22/0/1	20/0/3	**22/1/0**	**22/1/0**	19/3/1	**22/1/0**
G_SPEA II	21/1/1	**22/1/0**	21/1/1	17/0/6	20/1/2	**22/1/0**	17/3/3	18/3/2	15/5/3	18/1/4	17/3/3	12/4/7	20/1/2	12/1/10	**22/1/0**	20/2/1	**23/0/0**	22/0/1	**22/1/0**	**22/1/0**	19/4/0	**22/1/0**
PAES	0/0/23	0/1/22	0/1/22	0/1/22	8/5/10	1/0/22	3/2/18	7/1/15	11/3/9	14/0/9	6/3/14	0/4/19	0/0/23	1/5/17	3/0/20	0/0/23	2/1/20	0/1/22	1/1/21	5/1/17	2/2/19	4/2/17
G_PAES	1/1/21	2/0/21	2/2/19	2/1/20	2/6/15	3/3/17	6/0/17	10/2/11	14/6/3	16/2/5	14/1/8	12/3/8	3/0/20	14/2/7	4/1/18	3/2/18	4/0/19	2/2/19	3/0/20	10/7/6	4/4/15	6/3/14
UNSGA III	6/4/13	5/6/12	6/4/13	2/2/19	14/5/4	2/2/19	12/3/8	18/3/2	0/2/21	21/1/1	21/1/1	3/4/16	18/0/5	1/4/18	13/2/8	17/1/5	15/2/6	17/1/5	8/1/14	7/3/13	12/4/7	0/1/22
G_UNSGA III	13/1/9	11/1/11	12/2/9	5/2/16	14/5/4	16/3/4	11/4/8	**22/1/0**	1/3/19	**23/0/0**	**23/0/0**	12/3/8	19/1/3	14/2/7	20/0/3	19/1/3	17/3/3	21/0/2	9/3/11	10/6/7	17/1/5	3/1/19
NSGA II	15/1/7	13/2/8	16/0/7	6/6/11	**23/0/0**	4/2/17	15/3/5	18/3/2	11/4/8	20/1/2	17/3/3	4/4/15	22/0/1	3/5/15	14/5/4	20/2/1	20/1/2	17/1/5	17/0/6	12/6/5	2/7/14	13/2/8
G_NSGA II	**23/0/0**	**22/1/0**	21/1/1	6/6/11	22/0/1	20/0/3	15/3/5	22/1/0	11/4/8	20/2/1	21/1/1	18/2/3	**23/0/0**	14/2/7	**22/1/0**	**23/0/0**	18/3/2	**23/0/0**	19/2/2	20/1/2	13/2/8	13/2/8
RVGEA	7/3/13	5/6/12	6/4/13	5/5/13	1/2/20	7/3/13	21/2/0	9/0/14	12/8/3	7/2/14	6/2/15	10/2/11	11/1/11	8/4/11	13/3/7	7/3/13	6/3/14	14/1/8	13/3/7	0/0/23	19/3/1	18/1/4
G_RVGEA	10/1/12	5/6/12	10/1/12	12/2/9	1/5/17	8/2/13	**22/1/0**	12/2/9	**21/2/0**	10/1/12	9/1/13	**22/1/0**	14/2/7	**22/1/0**	18/1/4	13/1/9	6/3/14	14/1/8	18/0/5	1/3/19	**21/2/0**	18/2/3
CTAEA	13/1/9	15/2/6	12/2/9	5/7/11	0/0/23	11/4/8	21/1/1	13/1/9	14/5/4	7/2/14	7/2/14	7/2/14	11/2/10	1/6/16	11/1/11	7/4/12	11/0/12	11/0/12	0/0/23	2/2/19	0/0/23	16/1/6
G_CTAEA	17/3/3	19/0/4	17/1/5	18/0/5	6/5/12	15/2/6	12/8/3	15/1/7	14/6/3	10/1/12	11/2/10	**22/1/0**	14/2/7	17/0/6	16/2/5	13/1/9	12/2/9	12/0/11	13/3/7	11/7/5	19/4/0	16/1/6
NSGA III	2/2/19	0/1/22	2/2/19	0/1/22	2/4/17	0/0/23	3/2/18	7/1/15	3/2/18	4/3/16	3/2/18	0/4/19	5/1/17	0/0/23	4/1/18	3/2/18	2/1/20	1/1/21	1/1/21	2/2/19	1/3/19	2/7/14
G_NSGA III	17/3/3	13/4/6	6/4/13	8/4/11	1/5/17	7/1/15	1/1/21	2/2/19	6/3/14	12/1/10	14/1/8	16/2/5	9/1/13	21/0/2	0/2/21	10/2/11	9/1/13	8/2/13	6/1/16	20/1/2	10/2/11	6/3/14
MOEAD	5/0/18	3/0/20	5/0/18	3/1/19	9/5/9	2/3/18	15/5/3	5/1/17	4/2/17	2/0/21	2/0/21	4/5/14	4/0/19	1/8/14	8/1/14	3/2/18	5/0/18	9/1/13	8/4/11	1/1/21	1/1/21	11/1/11
G_MOEAD	6/3/14	4/1/18	12/3/8	6/5/12	6/4/13	16/3/4	8/1/14	10/1/12	9/2/12	12/1/10	12/1/10	19/1/3	9/1/13	18/2/3	10/0/13	6/0/17	12/2/9	13/0/10	9/3/11	5/1/17	11/3/9	11/1/11
IBEA_FC	12/0/11	13/2/8	14/1/8	7/7/9	2/4/17	8/2/13	3/2/18	5/1/17	8/2/13	0/1/22	0/1/22	5/4/14	5/1/17	7/4/12	8/1/14	7/4/12	12/2/9	7/0/16	12/3/8	7/2/14	4/5/14	0/1/22
G_IBEA_FC	21/1/1	20/1/2	**23/0/0**	15/1/7	14/3/6	15/4/4	7/0/16	11/2/10	10/10/3	0/1/22	0/1/22	21/0/2	12/1/10	**22/1/0**	16/2/5	15/1/7	15/4/4	8/1/14	14/2/7	18/1/4	14/2/7	2/1/20
MOPSO	6/4/13	5/6/12	6/5/12	20/2/1	7/7/9	11/4/8	9/1/13	0/0/23	0/3/20	4/3/16	3/2/18	0/2/21	1/1/21	1/10/12	0/1/22	1/1/21	6/4/13	4/2/17	4/1/18	11/7/5	4/4/15	20/1/2
G_MOPSO	6/5/12	5/6/12	6/4/13	20/2/1	7/10/6	11/4/8	8/2/13	1/0/22	0/3/20	3/1/19	3/2/18	1/7/15	1/1/21	4/7/12	1/1/21	1/1/21	6/3/14	4/2/17	4/1/18	11/7/5	4/4/15	19/2/2
SMPSO	1/3/19	4/7/12	0/1/22	19/1/3	10/4/9	2/4/17	0/1/22	2/2/19	6/2/15	3/3/17	6/4/13	15/2/6	7/1/15	18/2/3	6/0/17	8/4/11	0/1/22	0/4/19	6/1/16	7/3/13	7/2/14	4/2/17
G_SMPSO	2/2/19	5/7/11	2/2/19	19/3/1	9/5/9	11/4/8	0/2/21	2/2/19	5/3/15	5/4/14	11/1/11	17/2/4	7/1/15	18/2/3	7/0/16	7/5/11	0/1/22	2/4/17	9/4/10	8/4/11	10/3/10	6/3/14
GREA	15/1/7	15/2/6	17/1/5	12/2/9	16/3/4	21/0/2	12/3/8	15/2/6	**21/2/0**	15/1/7	16/0/7	1/4/18	14/2/7	2/7/14	11/1/11	15/1/7	15/2/6	16/0/7	19/2/2	11/6/6	10/6/7	10/0/13
G_GREA	17/3/3	20/1/2	19/1/3	**23/0/0**	16/3/4	**22/1/0**	11/1/11	16/1/6	**21/2/0**	15/2/6	17/3/3	10/5/8	17/0/6	5/5/13	13/3/7	17/1/5	17/3/3	19/0/4	19/2/2	14/5/4	17/1/5	13/2/8

TABLE III
FRIEDMAN TEST RANKING BASED ON IGD AT THE 50-TH ITERATION FOR 30 TIMES RUNS (100 POPULATION SIZE).

Algorithm	ZDT						WFG					DTLZ							IMOP				Average
	1	2	3	4	5	6	1	2	3	4	5	1	2	3	4	5	6	7	1	2	3	4	
SPEA II	5.467	6.633	4.433	9.367	3.667	9.367	5.533	4.667	6.800	5.967	5.833	13.77	4.033	13.27	3.267	4.033	1.900	3.933	1.733	1.733	5.767	1.933	5.595
G_SPEA II	2.400	**1.667**	2.800	7.500	4.100	**1.567**	5.033	4.433	6.000	5.733	5.500	8.700	3.533	12.53	2.033	3.133	**1.100**	1.767	**1.600**	**1.300**	3.533	**1.767**	**3.988**
PAES	23.70	23.50	23.20	23.33	13.57	22.50	20.57	16.67	11.43	9.833	16.07	21.23	23.77	20.13	21.63	23.73	21.87	23.00	22.30	17.57	20.93	18.47	19.95
G_PAES	21.90	22.00	21.27	20.70	17.77	19.83	19.00	12.67	7.533	6.833	9.000	9.833	21.47	9.100	19.17	20.47	19.97	21.30	20.17	11.27	17.57	17.07	16.63
UNSGA III	15.80	15.70	15.50	19.80	8.867	20.67	10.43	3.667	21.80	2.667	3.033	18.97	6.300	20.10	10.13	6.633	7.067	6.700	15.43	15.47	11.03	23.33	12.69
G_UNSGA III	11.10	12.57	11.50	17.53	9.000	6.333	10.70	2.800	20.90	**1.900**	**1.567**	9.733	4.267	9.367	4.200	4.100	5.667	3.467	13.33	11.13	8.167	20.50	9.083
NSGA II	8.233	10.63	8.433	14.67	**1.133**	19.20	8.033	4.267	10.50	3.767	5.200	18.33	2.100	18.33	7.567	2.933	4.567	6.333	7.400	9.767	18.30	9.933	9.074
G_NSGA II	**1.467**	2.133	2.367	14.57	2.700	4.500	7.833	**2.067**	10.87	2.900	2.767	5.400	**1.133**	8.567	**1.700**	**1.433**	5.167	**1.300**	4.533	4.100	10.37	10.03	4.905
RVGEA	15.27	16.27	16.13	16.23	20.27	15.17	2.000	14.77	8.700	16.27	17.17	13.67	12.73	14.50	9.567	15.43	16.10	10.10	9.700	23.03	3.500	5.467	13.27
G_RVGEA	14.40	15.60	13.43	11.17	18.97	15.00	**1.733**	11.30	3.433	12.80	14.20	**2.133**	8.867	**1.367**	5.600	11.03	16.23	9.633	6.333	20.87	**2.267**	4.833	10.05
CTAEA	11.07	8.667	12.03	15.50	22.67	12.17	3.767	10.50	8.067	16.23	16.07	16.67	11.90	19.43	11.10	14.77	12.70	13.27	23.33	20.67	22.03	8.400	14.14
G_CTAEA	5.233	6.067	6.700	5.933	15.93	7.700	8.233	8.900	6.867	13.20	12.13	3.067	9.367	6.833	7.533	11.07	12.10	12.47	10.77	9.233	2.833	7.167	8.606
NSGA III	21.37	23.23	21.17	23.33	18.37	23.77	20.13	17.00	19.27	18.40	19.37	21.47	18.20	23.00	19.60	20.13	21.93	22.13	21.60	20.70	20.30	19.00	20.61
G_NSGA III	7.100	10.23	15.33	14.33	18.57	16.27	21.70	20.20	16.17	11.77	9.333	7.067	14.57	3.567	22.70	13.53	14.33	15.67	15.97	4.100	12.67	16.97	13.73
MOEAD	19.13	20.60	18.77	19.00	12.90	20.27	7.600	17.97	18.90	21.90	21.53	18.27	19.40	18.37	15.73	20.73	19.10	14.47	13.73	22.43	21.50	11.90	17.92
G_MOEAD	15.80	18.00	10.90	15.13	16.27	6.800	14.77	13.37	14.10	11.93	11.93	4.833	14.73	5.367	13.10	17.13	11.60	10.53	13.33	18.33	11.23	12.23	12.79
IBEA_FC	11.87	10.53	9.567	13.00	19.00	14.53	19.87	18.63	14.57	23.33	23.30	17.73	17.83	15.20	15.13	14.50	12.23	17.20	10.90	15.93	16.27	23.17	16.10
G_IBEA_FC	2.133	2.867	**1.233**	8.833	9.667	7.267	16.47	11.97	9.767	23.67	23.70	3.367	11.27	1.833	7.900	8.267	6.800	16.07	8.767	6.767	9.167	21.47	9.965
MOPSO	15.33	15.67	15.10	3.033	14.40	11.17	14.30	23.57	21.50	19.83	19.73	20.60	22.67	17.23	23.10	22.07	16.03	18.77	19.67	9.867	18.17	4.367	16.57
G_MOPSO	13.73	14.90	14.17	**2.767**	12.97	11.00	13.90	23.07	21.47	18.77	19.73	19.20	22.10	16.00	22.00	21.83	15.57	19.10	18.87	9.533	17.57	4.000	16.08
SMPSO	21.87	16.53	22.63	5.700	12.03	18.37	23.20	21.33	17.47	18.93	16.17	8.733	17.13	5.600	17.97	13.90	22.73	20.80	17.83	14.10	16.13	18.07	16.69
G_SMPSO	20.97	14.97	21.60	3.933	12.43	11.43	22.37	20.70	17.10	17.60	13.50	6.700	16.53	5.433	17.23	14.60	22.47	19.50	13.17	12.77	12.67	16.43	15.19
GREA	8.800	7.467	6.967	11.87	7.200	3.367	11.00	8.100	**3.333**	8.067	7.233	19.70	8.700	18.13	12.33	7.767	7.100	7.467	4.900	10.50	11.03	13.93	9.317
G_GREA	5.867	3.567	4.767	**2.767**	7.567	1.733	11.83	7.400	3.467	7.700	5.933	10.77	7.400	16.73	9.700	6.767	5.667	5.033	4.633	8.033	7.000	9.567	6.995

built on different algorithms and different problems, and the optimal scales of different algorithms and problems being different, additional considerations need to be given to parameter settings during the application of meta-heuristics, which is also a difficult point in meta-heuristic research. The combination of GMHA and hybrid multi-growth routes actually improves the statistical performance of meta-heuristic algorithms from a statistical perspective by increasing the adaptation probability.

V. CONCLUSION AND FUTURE DIRECTION

Due to the superior performance and irreplaceability of meta-heuristic algorithms (MHAs) in solving MOPs, especially for Cloud scheduling, a universal strategy or framework that can enhance the overall performance of various MHAs is of profound significance and also a highly challenging topic.

In response to this challenge, the main contributions of this paper include reconstructing a universal MHA framework by integrating the common features of various MHAs, proposing the growable meta-heuristic algorithm framework (GMHA) with the hybrid multi-growth routes strategy combining several proposed general growth routes, and concretely instantiating multiple growable meta-heuristic algorithms, which practically expands the capability boundary of MHAs in MOPs.

Integrating the advantages of the universal meta-heuristic framework and hybrid multi-growth routes strategy, GMHA possesses universality for various basic MHAs and MOPs. Extensive experiments have verified this universality from multiple perspectives. From statistical test, GMHA has a probability of 90.15% to improve the performance of various MHAs in various MOP benchmarks. Experiments in Cloud scheduling shows that the average convergence speed of GMHA is 2.116× of the corresponding MHA, achieving reductions of 25.05% in total energy consumption, 4.94% in the maximum energy consumption of server nodes, and 35.12% in the sum of the standard deviations of utilizations.

As an exploration of MHA framework, GMHA demonstrates its potential. However, during the exploration of the universal framework, this paper only implements the fixed meta-heuristic strategy and growth parameters, as well as uses the random selector based on uniform distribution to determine the growth routes, hence showing limitations and instability in certain problems or scenarios. In the future, part of our work is to study how to dynamically set the configurations of growth parameters, growth routes and basic meta-heuristic search strategies based on the context during the optimization process for different problems and algorithms in services-oriented distributed computing. Furthermore, theoretical modeling and analysis of GMHA's performance are also worth exploring.

Acknowledgments

This research is partially supported by the Fundamental Research Funds for the Central Universities with Grant ID 2682025CX127, and the National Natural Science Foundation of China with Grant ID 62471411.

References

[1] J. Liu, X. Li, F. Tang *et al.*, "An adaptive and interpretable congestion control service based on multi-objective reinforcement learning," *IEEE Trans. Serv. Comput.*, vol. 18, no. 2, pp. 586–602, 2025.

[2] T. H. B. Huy, H. T. Doan, V. N. Dieu *et al.*, "Multi-objective optimal power flow of thermal-wind-solar power system using an adaptive geometry estimation based multi-objective differential evolution," *Appl. Soft Comput.*, vol. 149, no. Part A, p. 110977, 2024.

[3] Y. Hu, J. Zou, J. Zheng *et al.*, "A new framework of change response for dynamic multi-objective optimization," *Expert Syst. Appl.*, vol. 248, p. 123344, 2024.

[4] Z. N. Ansari and S. D. Daxini, "A state-of-the-art review on meta-heuristics application in remanufacturing," *Archives Of Computational Methods In Engineering*, vol. 29, no. 1, pp. 427–470, 2022.

[5] A. Kumar, S. Das, and R. Mallipeddi, "An efficient differential grouping algorithm for large-scale global optimization," *IEEE Trans. Evol. Comput.*, vol. 28, no. 1, pp. 32–46, 2024.

[6] F. J. Kaleibar, M. St-Hilaire, and M. Barati, "A customized genetic algorithm for sla-aware service provisioning in infrastructure-less vehicular cloud networks," *IEEE Trans. Serv. Comput.*, vol. 18, no. 2, pp. 499–512, 2025.

[7] G. Zhou, Y. Xie, H. Lan *et al.*, "Information interaction and partial growth-based multi-population growable genetic algorithm for multi-dimensional resources utilization optimization of cloud computing," *Swarm Evol. Comput.*, vol. 87, p. 101575, 2024.

[8] J. Zhou, L. Gao, S. Rao, and Y. Li, "Scheduling constrained cloud workflow tasks via evolutionary multitasking optimization with adaptive knowledge transfer," *IEEE Trans. Serv. Comput.*, vol. 17, no. 6, pp. 4254–4266, 2024.

[9] K. Rajwar, K. Deep, and S. Das, "An exhaustive review of the metaheuristic algorithms for search and optimization: taxonomy, applications, and open challenges," *Artif. Intell. Rev.*, vol. 56, no. 11, pp. 13 187–13 257, 2023.

[10] M. Kaveh and M. S. Mesgari, "Application of meta-heuristic algorithms for training neural networks and deep learning architectures: A comprehensive review," *Neural Process. Lett.*, vol. 55, no. 4, pp. 4519–4622, 2023.

[11] B. Yang, J. Wang, X. Zhang *et al.*, "Comprehensive overview of metaheuristic algorithm applications on pv cell parameter identification," *Energy Conversion and Management*, vol. 208, p. 112595, 2020.

[12] S. John, "Model-driven optimization with a focus on the effectiveness and efficiency of evolutionary algorithms," Ph.D. dissertation, University of Marburg, Germany, 2024.

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

[14] S. Karami, S. Azizi, and F. Ahmadizar, "A bi-objective workflow scheduling in virtualized fog-cloud computing using NSGA-II with semi-greedy initialization," *Appl. Soft Comput.*, vol. 151, p. 111142, 2024.

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

[16] H. Seada and K. Deb, "U-NSGA-III: A unified evolutionary optimization procedure for single, multiple, and many objectives: Proof-of-principle results," in *Evolutionary Multi-Criterion Optimization - 8th International Conference. Proceedings, Part II*, ser. Lecture Notes in Computer Science, vol. 9019. Springer, 2015, pp. 34–49.

[17] K. Li, R. Chen, G. Fu, and X. Yao, "Two-archive evolutionary algorithm for constrained multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 23, no. 2, pp. 303–315, 2019.

[18] Y. Li, W. Li, S. Li, and Y. Zhao, "A performance indicator-based evolutionary algorithm for expensive high-dimensional multi-/many-objective optimization," *Inf. Sci.*, vol. 678, p. 121045, 2024.

[19] V. K. Maurya and S. J. Nanda, "Time-varying multi-objective smart home appliances scheduling using fuzzy adaptive dynamic SPEA2 algorithm," *Eng. Appl. Artif. Intell.*, vol. 121, p. 105944, 2023.

[20] J. Tang, H. Duan, and S. Lao, "Swarm intelligence algorithms for multiple unmanned aerial vehicles collaboration: a comprehensive review," *Artif. Intell. Rev.*, vol. 56, no. 5, pp. 4295–4327, 2023.

[21] S. Z. Martínez and C. A. C. Coello, "A multi-objective particle swarm optimizer based on decomposition," in *13th Annual Genetic and Evolutionary Computation Conference, GECCO 2011*. ACM, 2011, pp. 69–76.

[22] A. J. Nebro, J. J. Durillo, J. García-Nieto *et al.*, "SMPSO: A new pso-based metaheuristic for multi-objective optimization," in *2009 IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making, MCDM 2009, Nashville, TN, USA, March 30 - April 2, 2009*. IEEE, 2009, pp. 66–73.

[23] D. Feng, Y. Li, J. Liu, and Y. Liu, "A particle swarm optimization algorithm based on modified crowding distance for multimodal multi-objective problems," *Appl. Soft Comput.*, vol. 152, p. 111280, 2024.

[24] J. Bai and H. Liu, "Multi-objective artificial bee algorithm based on decomposition by PBI method," *Appl. Intell.*, vol. 45, no. 4, pp. 976–991, 2016.

[25] B. Zhang, A. Che, and Y. Wang, "Grid-based artificial bee colony algorithm for multi-objective job shop scheduling with manual loading and unloading tasks," *Expert Syst. Appl.*, vol. 245, p. 123011, 2024.

[26] S. E. Comert and H. R. Yazgan, "A new approach based on hybrid ant colony optimization-artificial bee colony algorithm for multi-objective electric vehicle routing problems," *Eng. Appl. Artif. Intell.*, vol. 123, no. Part B, p. 106375, 2023.

[27] Y. Kang, X. Yang, B. Pu *et al.*, "HWOA: an intelligent hybrid whale optimization algorithm for multi-objective task selection strategy in edge cloud computing system," *World Wide Web*, vol. 25, no. 5, pp. 2265–2295, 2022.

[28] M. Zhang, J. Wang, Y. Liu *et al.*, "Multi-objective optimization algorithm based on clustering guided binary equilibrium optimizer and NSGA-III to solve high-dimensional feature selection problem," *Inf. Sci.*, vol. 648, p. 119638, 2023.

[29] M. Abdel-Basset, R. Mohamed, S. Mirjalili *et al.*, "MOEO-EED: A multi-objective equilibrium optimizer with exploration-exploitation dominance strategy," *Knowl. Based Syst.*, vol. 214, p. 106717, 2021.

[30] Y. Huang, B. Sheng, G. Fu *et al.*, "Multi-objective simulated annealing algorithm for robotic mixed-model two-sided assembly line balancing with setup times and multiple constraints," *Appl. Soft Comput.*, vol. 156, p. 111507, 2024.

[31] M. Ayar, A. Isazadeh, F. S. Gharehchopogh, and M. Seyedi, "NSICA: multi-objective imperialist competitive algorithm for feature selection in arrhythmia diagnosis," *Comput. Biol. Medicine*, vol. 161, p. 107025, 2023.

[32] W. Lin, W. Wu, and L. He, "An on-line virtual machine consolidation strategy for dual improvement in performance and energy conservation

of server clusters in cloud data centers," *IEEE Trans. Serv. Comput.*, vol. 15, no. 2, pp. 766–777, 2022.
[33] Q. Shen, B. Hu, and E. Xia, "Cuckoo search-enabled task scheduling and cache updating in vehicular edge-fog computing," *IEEE Trans. Veh. Technol.*, vol. 74, no. 6, pp. 9656–9670, 2025.
[34] D. A. Monge, E. Pacini, C. Mateos *et al.*, "CMI: an online multi-objective genetic autoscaler for scientific and engineering workflows in cloud infrastructures with unreliable virtual machines," *J. Netw. Comput. Appl.*, vol. 149, 2020.
[35] B. H. Abed-alguni and N. A. Alawad, "Distributed grey wolf optimizer for scheduling of workflow applications in cloud environments," *Appl. Soft Comput.*, vol. 102, p. 107113, 2021.
[36] S. Mehta, P. Kaur, and P. Agarwal, "Improved whale optimization variants for sla-compliant placement of virtual machines in cloud data centers," *Multim. Tools Appl.*, vol. 83, no. 1, pp. 149–171, 2024.
[37] Z. Sun, Y. Mei *et al.*, "Multi-tree genetic programming hyper-heuristic for dynamic flexible workflow scheduling in multi-clouds," *IEEE Trans. Serv. Comput.*, vol. 17, no. 5, pp. 2687–2703, 2024.
[38] Y. Yang, H. Shen, and H. Tian, "Scheduling workflow tasks with unknown task execution time by combining machine-learning and greedy-optimization," *IEEE Trans. Serv. Comput.*, vol. 17, no. 3, pp. 1181–1195, 2024.
[39] Z. Xiao, Q. Qiu, L. Li *et al.*, "An efficient service-aware virtual machine scheduling approach based on multi-objective evolutionary algorithm," *IEEE Trans. Serv. Comput.*, vol. 17, no. 5, pp. 2027–2040, 2024.
[40] H. Qiu, X. Xia, Y. Li, and X. Deng, "A dynamic multipopulation genetic algorithm for multiobjective workflow scheduling based on the longest common sequence," *Swarm Evol. Comput.*, vol. 78, p. 101291, 2023.
[41] L. Yin and Z. Sun, "Distributed multi-objective grey wolf optimizer for distributed multi-objective economic dispatch of multi-area interconnected power systems," *Appl. Soft Comput.*, vol. 117, p. 108345, 2022.
[42] B. H. Abed-alguni and D. J. Paul, "Island-based cuckoo search with elite opposition-based learning and multiple mutation methods for solving optimization problems," *Soft Comput.*, vol. 26, no. 7, pp. 3293–3312, 2022.
[43] J. D. Knowles and D. Corne, "The pareto archived evolution strategy: a new baseline algorithm for pareto multiobjective optimisation," in *Proceedings of the 1999 Congress on Evolutionary Computation, CEC 1999*. IEEE, 1999, pp. 98–105.
[44] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: Empirical results," *Evol. Comput.*, vol. 8, no. 2, pp. 173–195, 2000.
[45] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable test problems for evolutionary multiobjective optimization," in *Evolutionary Multiobjective Optimization*, ser. Advanced Information and Knowledge Processing. Springer, 2005, pp. 105–145.
[46] S. Liu, Q. Lin, K. Wong *et al.*, "Evolutionary large-scale multiobjective optimization: Benchmarks and algorithms," *IEEE Trans. Evol. Comput.*, vol. 27, no. 3, pp. 401–415, 2023.
[47] Y. Tian, R. Cheng, X. Zhang *et al.*, "Diversity assessment of multi-objective evolutionary algorithms: Performance metric and benchmark problems [research frontier]," *IEEE Comput. Intell. Mag.*, vol. 14, no. 3, pp. 61–74, 2019.
[48] J. Blank and K. Deb, "Pymoo: Multi-objective optimization in python," *IEEE Access*, vol. 8, pp. 89 497–89 509, 2020.

Guangyao Zhou received Bachelor's degree and Master's degree from Tianjin University, China. He received Ph.D degree from University of Electronic Science and Technology of China, China. He received the excellent doctoral dissertation award of University of Electronic Science and Technology of China. He is now an assistant professor at the School of Computing and Artificial Intelligence, Southwest Jiaotong University, China. His research interests include scheduling algorithms in Cloud Computing or Edge Computing, image recognition especially facial expression recognition, parallel training and reasoning of large-scale model and evolution algorithms.

Minxian Xu received PhD degree from the University of Melbourne. He is now an associate professor at Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences. His research interests include resource scheduling and optimization in cloud computing. He has co-authored 60+ peer-reviewed papers published in prominent international journals and conferences. His PhD thesis was awarded the 2019 IEEE TCSC Outstanding Ph.D. Dissertation Award. He was also awarded the 2023 IEEE TCSC Award for Excellence (Early Career Award).

Jiashu Zhang is now a professor at School of Computing and Artificial Intelligence, Southwest Jiaotong University, China. He is currently the Director of the Sichuan Province Key Laboratory of Signal and Information Process ing. He has coauthored more than 200 journal and conference papers. His research interests include nonlinear and adaptive signal processing, biometric security and privacy, intelligent computing, information forensic and data hiding. He is a senior member of the Chinese Institute of Electronics and the Chinese Institute of Communications.

Rajkumar Buyya is a Redmond Barry Distinguished Professor and Director of the Cloud Computing and Distributed Systems (CLOUDS) Laboratory at the University of Melbourne, Australia. He served as a Future Fellow of the Australian Research Council during 2012-2016. He has authored over 750 publications and seven text books. He is one of the highly cited authors in computer science and software engineering worldwide (h-index=172, gindex=384, 159800+ citations). He is recognized as a "Web of Science Highly Cited Researcher" for six consecutive years since 2016, and Scopus Researcher of the Year 2017 with Excellence in Innovative Research Award by Elsevier for his outstanding contributions to Cloud Computing and distributed systems.