In: Information Sciences

An adaptive multi-population differential artificial bee colony algorithm for many-objective service composition in cloud manufacturing

Jiajun Zhou[a], Xifan Yao[a,*], Yingzi Lin[b], Felix T.S. Chan[c], Yun Li[d]

[a] School of Mechanical and Automotive Engineering, South China University of Technology, Guangzhou 510640, Guangdong, China
[b] Intelligent Human Machine Systems Lab, Department of Mechanical and Industrial Engineering, Northeastern University, Boston, MA 02115, USA
[c] Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University, Hung Hom, Hong Kong
[d] Faculty of Engineering, University of Strathclyde, Glasgow G1 1XQ, UK

A B S T R A C T

Several conflicting criteria must be optimized simultaneously during the service composition and optimal selection (SCOS) in cloud manufacturing, among which tradeoff optimization regarding the quality of the composite services is a key issue in successful implementation of manufacturing tasks. This study improves the artificial bee colony (ABC) algorithm by introducing a synergetic mechanism for food source perturbation, a new diversity maintenance strategy, and a novel computing resources allocation scheme to handle complicated many-objective SCOS problems. Specifically, differential evolution (DE) operators with distinct search behaviors are integrated into the ABC updating equation to enhance the level of information exchange between the foraging bees, and the control parameters for reproduction operators are adapted independently. Meanwhile, a scalarization based approach with active diversity promotion is used to enhance the selection pressure. In this proposal, multiple size adjustable subpopulations evolve with distinct reproduction operators according to the utility of the generating offspring so that more computational resources will be allocated to the better performing reproduction operators. Experiments for addressing benchmark test instances and SCOS problems indicate that the proposed algorithm has a competitive performance and scalability behavior compared with contesting algorithms.

1. Introduction

Cloud manufacturing (CMfg) refers to a service-oriented networked manufacturing model in which service consumers are enabled to configure, select, and utilize resources on demand so as to complete customized manufacturing tasks [9]. Adapted from the cloud computing model and Internet of things (IoT) into the field of manufacturing, CMfg was first coined by Li et al. [17] and is gaining significant attention from both academia and industry. CMfg aims at connecting consumers with small and medium-sized enterprises or individual engineers to form temporary, reconfigurable production lines, providing consumers with capable and qualified design and manufacturing services in a virtual community.

* Corresponding author.
E-mail addresses: mezjj@foxmail.com (J. Zhou), mexfyao@scut.edu.cn (X. Yao).

To facilitate the access and utilization of resources in heterogeneous environments, manufacturing resources or capacities in CMfg are virtualized and delivered in the form of services. However, due to the functional limitation of a single service, the complexity of a manufacturing task calls for a composite service in which multiple isolated resources are organized into a set of interacting services with more powerful functions. On the other hand, the increasing diversity and complexity of customers' individualization requirements has resulted in the difficulty for a single functional service to complete a user's task. From the perspective of cost and efficiency, a specialized service for each task means relatively high cost and slow response. Therefore, assembling modular services into an aggregated one to meet a customer's requirement will strengthen the flexibility and agility for CMfg. For a complicated manufacturing task, the CMfg system decomposes it into several subtasks and searches all qualified services for each subtask to form a candidate service pool, albeit with different QoS (quality of service). Then, the system selects one service from each service pool to generate a composite service with the given multi-objective optimization (e.g., multiple conflicting QoS properties need to be optimized simultaneously), and such a process is called service composition optimal selection (SCOS). Obviously, with the mass of services available on the pool, SCOS belongs to the set of NP-hard multi-objective optimization problems with characteristics of nonlinear objective functions, combinatorial explosion, and high-dimensional decision space.

Various approaches have been proposed for solving such problems (e.g., [31,33]). Due to the combinatorial explosion, exact approaches or exhaustive search algorithms for SCOS are computationally expensive or suffer from finding an optimal solution within a polynomial time. To address this issue, scholars have developed many evolutionary algorithms such as the genetic algorithm (GA) [31,33] and particle swarm optimization (PSO) [34]. In these studies, SCOS in different kinds of scenarios for achieving specific goals is discussed. However, almost all of them transform the multi-objective problems (MOPs) into single objective problems (SOPs) by weighted combination. This is somewhat rigid in terms of flexibility. Due to the conflicting nature of multiple objectives in MOPs, a single solution, therefore, is not available. Instead, a set of tradeoff solutions, called a Pareto optimal set, is generally preferred practically. Users can select one of them according to their preference. Therefore, providing multiple alternatives with Pareto optimality in a single run by using multi-objective evolutionary algorithms (MOEAs) is becoming more desirable. Compared with SOPs, MOPs are more complex, whose optimization target is to achieve the best tradeoff between multiple objectives, and the algorithms may need more interactions to form a hybrid optimizer with powerful searching ability.

The artificial bee colony (ABC) algorithm is a fairly new swarm intelligent algorithm inspired by honeybee foraging behavior, and was first developed by Karaboga and Basturk [14]. With fewer parameters, ABC is relatively efficient, robust and successful in solving the real world problems [30]. The basic version of ABC is very efficient in tackling SOPs [15]. However, its performance is not effective in MOP and many-objective problem contexts. This is mainly due to the inefficient food source perturbation in the case of a high dimensional objective space. To overcome such drawbacks, we introduce several promising DE mutation strategies to strengthen the honeybee foraging behavior, and adjust the operator control parameters for good offspring reproduction. In addition, the quality-indicator based fitness assignment approach is employed to identify prominent search regions in the ABC onlookers stage. Furthermore, a well-maintained external archive tailored for many-objective contexts is employed to guide the collaboration of search operators.

The rest of this paper is organized as follows. Section 2 reviews the relevant work concerning service composition and multi-objective evolutionary computing. In Section 3, the SCOS problem and its optimization model are described. Section 4 explains the newly proposed algorithm and its key components. In Section 5, comparison experiments on a wide range of benchmark problems are discussed and analyzed. The proposed algorithm is then applied to tackle SCOS problems in Section 6. Finally, conclusions are given in Section 7.

2. Related work

From the standpoint of service accessibilities and resource allocation efficiency, designing all the possible required services for a task in the real world is extremely difficult and is somewhat rigid in terms of flexibility, particularly for a customer's personalized demand. Providing simple and single fundamental services that can be configured, selected, and collaborated for customized scenarios make more sense since such approaches are more agile, scalable and elastic. As a result, how to select appropriate services from multiple candidates with different QoS properties to generate a composite service while guarantying the best QoS of the result must be investigated and addressed. Unfortunately, such a process is characterized with NP-hard complexity due to the exuberant growth of services offered in the cloud, and most of the existing methods meet a performance bottleneck when handling the composition.

Cloud computing further accelerates the increase of available services on the Internet, and QoS has drawn more attention due to many candidates meeting the same functional requirement, albeit with distinct QoS. As a result, it is a challenge for composition systems to choose an optimal set of required services from different service pools efficiently while guaranteeing the overall QoS. Composition techniques for cloud services first appeared in 2009 [13], and afterwards, substantial academic efforts were made in this field. With the capabilities of handling complex situations, gradient-free mechanisms, and obtaining near-optimal solutions within a short period of time, evolutionary algorithms have attracted much attention in recent years, becoming a popular approach in the literature for solving SCOS problems. In [33], GA and its variants were used in partner selection in virtual manufacturing. In [34], an enhanced PSO was developed for SCOS in a manufacturing grid. Huang et al. [11] used a new chaos control optimal algorithm with better population diversity to address the SCOS problem, which performed better than the chaotic GA. In [32], a parallel chaos optimizer named FC-PACO-RM was introduced to handle

SCOS in CMfg. Laili et al. [16] analyzed the characteristics of resource services in the private cloud, and constructed a dual scheduling model for manufacturing service selection.

Generally, the aforementioned QoS-aware SCOS approaches show distinct characteristics and have been applied in different scenarios. However, there is still room for adaptability and efficiency improvement in these approaches. In fact, due to the complexity of the CMfg environment and the large scale of available candidate services, current methods are undesirable to some extent and suffer from searching space explosion, difficulties such as loss of diversity and local convergence associated with the evolutionary process. Combining the merits of several algorithms is an effective way, which, however, often leads to the difficulty in determining an appropriate collaborative manner. Moreover, from the aspect of the objective dimension, most existing approaches try to create a utility function by aggregating multiple objectives together, which essentially transforms the multi-objective service composition into single objective optimization. In practice, whether or not the aggregation functions can truly reflect the importance of different QoS properties remains doubtful. Another drawback of the aggregation-based method concerns the inherent characteristics of the aggregation function. For instance, weighted sum aggregation is incapable of finding non-convex parts of the PF (Pareto Front). Thus, to provide diversified alternatives for users, a powerful algorithm for SCOS in a many-objective context is required.

ABC is a stochastic search algorithm based on the intelligent swarm behavior of bees, and has been applied for solving real word optimization problems [30,36] having great computational difficulties. In recent work, several modifications were made for the original ABC as well as hybridization with other meta-heuristic methods [36,48,49]. Examination of reliable publications shows that ABC potentially outperforms other well-known meta-heuristic approaches such as GA and PSO [15]. Nonetheless, limited efforts have been devoted to SCOS in CMfg. Indeed, several existing hybrid algorithms try to address SCOS but they only focus on one objective by using a scalarization-based approach [49] or two representative objectives by reduction [31,48]. In practice, the objective reduction strategy can be of limited use because it is hard to satisfy customers' many diverse preferences in focusing on different QoS aspects, which often involve four or more objectives. This phenomenon gives rise to a new term, formally known as many-objective optimization problems (MaOPs). MaOPs pose great challenges to existing Pareto-based multi-objective evolutionary algorithms (MOEAs) such as NSGA-II [6] and SPEA2 [50], as their Pareto-dominance-based selection pressure is severely degraded with the increasing number of objectives. As a result, Pareto-based MOEAs suffer from slow convergence to the PF. On the contrary, MOEA based on decomposition (MOEA/D) divides a complex MOP into a number of SOPs and solves them in a collaborative manner, which facilitates convergence by the scalarization-based approach. However, a potential drawback of MOEA/D is that high convergence pressure may lead the algorithm to converge on some narrow parts of PF when the population diversity is not well maintained. In recent years, a number of many-objective evolutionary algorithms (MaOEAs) have been suggested. Based on different selection criteria, MaOEAs can be roughly categorized into five types:

(1) Modified-Pareto-dominance-based approaches: referring to those overcoming the ineffectiveness of Pareto optimality caused by the curse of dimensionality, including θ-dominance [41], fuzzy Pareto dominance [10] and grid dominance [40].
(2) Decomposition-based approaches: referring to those using the framework of MOEA/D (e.g., MOEA/D-DU [42], EFR-RR [42], RPEA [26] and θ-DEA [41]) or MOEA/D-M2M [24] with an enhanced replacement strategy or different ranking schemes.
(3) Indicator-based approaches: referring to those utilizing various performance indicators like IGD-indicator [35], HV indicator [1], and R2 indicator [7] to evaluate the quality of solutions, such as AR-MOEA [35], HypE [1], and MOMBI-II [7].
(4) Diversity-based approaches: referring to those attempting to alleviate the loss of selection pressure through improved diversity management schemes, e.g., SPEA2 + SDE with shift-based density estimation [23], NSGA-III with niche-preservation operator [5], MaOEA-DDFC based on directional diversity and favorable convergence [4], VaEA with vector angle based diversity promotion [39], SPEA/R with reference direction-based density estimator [12], and 1by1EA with a distribution indicator [25].
(5) Hybrid approaches: referring to those adopting hybrid selection mechanisms to emphasize both approximation and diversity, including MaOEAs based on the combination of Pareto dominance- and decomposition-based selection mechanisms (e.g. MOEA/DD [20], and DPP [22]), and MaOEAs based on the combination of Pareto dominance- and indicator-based selection mechanisms (e.g., P-SEA [8] and Two-Arch2 [37]).

Motivated by the success of multiple selection criteria and the merit of ABC, we suggest a new optimizer, whose selection criteria, based on decomposition, indicator and pareto dominance, are merged into the ABC framework with a complementary effect. To be specific, decomposition based selection is used to enhance the selection pressure towards the true PF. The synergy of pareto- and decomposition-based approaches is employed to maintain the external archive for robust searching. The Indicator based approach is adopted to identify the potential search regions. As for reproduction, multiple different DE operators are employed in an adaptive manner. This proposed hybrid algorithm follows the identical framework of ABC while introducing multiple reproduction operators and a synergy of distinct selection criteria. Multi-populations with varying size are designed to adjust computational resources between operators automatically as evolution proceeds.

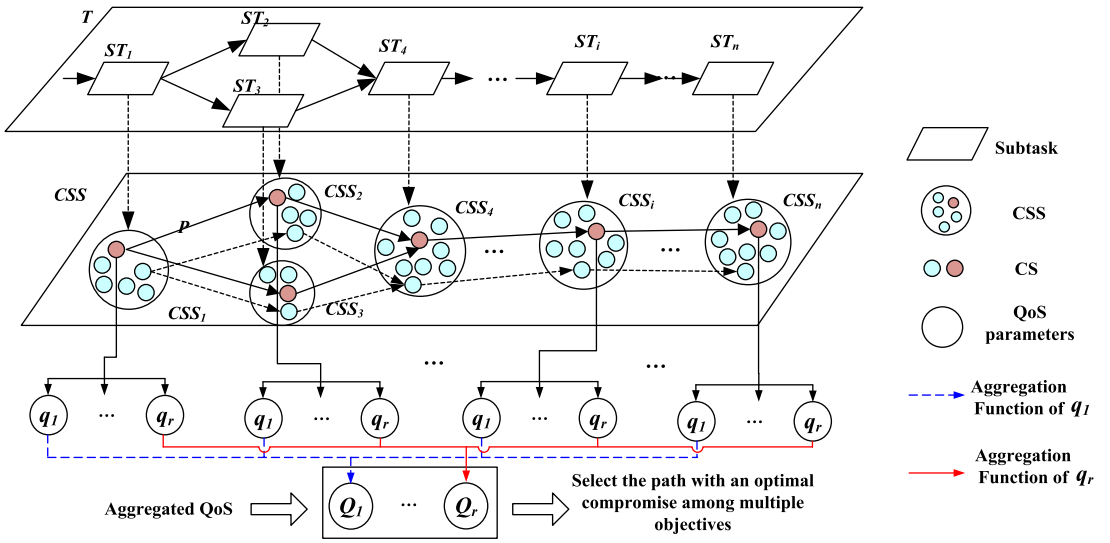

Fig. 1. Overview of SCOS procedure.

Table 1
Aggregation QoS evaluation.

Structure	Time	Cost	Reliability	Availability	Reputation
Sequence	$Q_1(seq) = \sum_{i=1}^{n} q_1(CS_i)$	$Q_2(seq) = \sum_{i=1}^{n} q_2(CS_i)$	$Q_3(seq) = \prod_{i=1}^{n} q_3(CS_i)$	$Q_4(seq) = \prod_{i=1}^{n} q_4(CS_i)$	$Q_5(seq) = \sum_{i=1}^{n} q_5(CS_i)/n$
Parallel	$Q_1(par) = \max(q_1(CS_i))$	$Q_2(par) = \sum_{i=1}^{n} q_2(CS_i)$	$Q_3(par) = \prod_{i=1}^{n} q_3(CS_i)$	$Q_4(par) = \prod_{i=1}^{n} q_4(CS_i)$	$Q_5(par) = \sum_{i=1}^{n} q_5(CS_i)/n$
Selective	$Q_1(sel) = \sum_{i=1}^{n} (q_1(CS_i) * \lambda_i)$	$Q_2(sel) = \sum_{i=1}^{n} (q_2(CS_i) * \lambda_i)$	$Q_3(sel) = \sum_{i=1}^{n} (q_3(CS_i) * \lambda_i)$	$Q_4(sel) = \sum_{i=1}^{n} (q_4(CS_i) * \lambda_i)$	$Q_5(sel) = \sum_{i=1}^{n} (q_5(CS_i) * \lambda_i)$
Circular	$Q_1(cir) = \theta * \sum_{i=1}^{n} q_1(CS_i)$	$Q_2(cir) = \theta * \sum_{i=1}^{n} q_2(CS_i)$	$Q_3(cir) = \prod_{i=1}^{n} q_3(CS_i)$	$Q_4(cir) = \prod_{i=1}^{n} q_4(CS_i)$	$Q_5(cir) = \sum_{i=1}^{n} q_5(CS_i)/n$

3. Problem statement

A complicated manufacturing task in CMfg can be represented by the directed acyclic diagram (DAG) with flow-dependent structures (e.g., sequence, parallel, selective, circular), in which the nodes, numbered from 1 to n, represent the decomposed subtasks, and the directed acyclic arrows denote the dependency relationships among the subtasks. Multiple candidate services (*CSs*) are generated according to the functional requirements of each subtask, i.e., a set of services with identical functionality but different non-functional properties forms the corresponding cloud service set (*CSS*). Formally, all the *CSs* meeting the requirements of the ith subtask ST_i are denoted as CSS_i. The *CSS* can be generated by semantic matching of the service functionality between supply and demand . Each *CS* is associated with several QoS parameters. Fig. 1 presents the procedure of SCOS along with its architecture layers and the optimal selection schema. Theoretically, a composite manufacturing service (*CMS*) consists of multiple CSs respectively selected from different *CSSs*. If the size of available *CSs* for ST_i is k_i, where $i \in \{1, 2, \cdots, n\}$, there will be $\prod_{i=1}^{n} k_i$ possible compositions, and the aim of SCOS is to find a *CMS* with the best QoS from all possible ones.

In practice, QoS consists of several non-functional metrics focusing on different aspects (e.g., time, cost, reliability, availability and reputation), so it is hard to achieve a solution with the best state in all metrics simultaneously. Instead, there exists a set of tradeoff options, known as Pareto-optimal solutions, where an improvement of one metric may result in deterioration of the others. For instance, time and cost can be clearly opposed, and improving the time would probably imply a deterioration in cost. Taking the importance of measurability into consideration, this paper focuses on five representative QoS criteria shown in Table 1, with the specific meaning of these QoS criteria explained in [11,32]. Interested readers are referred to [13] for more QoS criteria in cloud-based systems.

Given the QoS criteria mentioned above, a QoS set of component services can be expressed as $\boldsymbol{q} = \{q_1, q_2, q_3, q_4, q_5\}$, a QoS set of *CMS* is described by $\mathbf{Q} = \{Q_1, Q_2, Q_3, Q_4, Q_5\}$, and the QoS values of a *CMS* are determined by the QoS values of its component services as well as the interconnection structure among component services.

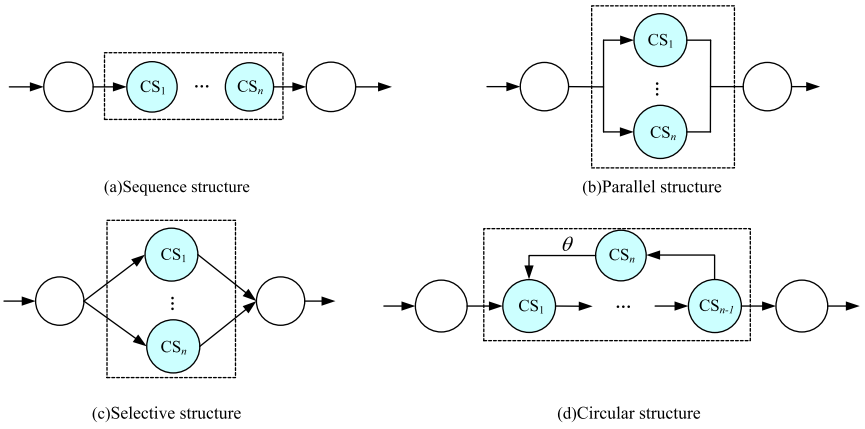

Fig. 2. Four basic composite structures.

In general, there are four basic composite structures: sequence, parallel, selective and circular, as shown in Fig. 2, and detailed formulas for aggregation QoS criteria values of composite services with different structures are presented in Table 1. Based on the formulas in Table 1, the parallel, selective and circular structures can be transformed into sequence structures [34]. Note that the scale of aggregated QoS criteria's values could be very different, which will seriously affect the scalarization function in decomposition based MOEAs. To avoid this issue, all the aggregated QoS values are normalized in the same range of [0, 1]. Specifically, the QoS criteria can be classified into positive and negative ones. For positive criteria, larger values mean better performance, while for negative ones, smaller values mean better performance. The normalization formulas for positive and negative criteria are shown in equations below, respectively.

$$\tilde{Q}_r(CMS) = \begin{cases} \frac{ag\left(q_r(CS_i^{j_i})\right) - ag\left(\overset{\min}{q_r}(CSS_i)\right)}{ag\left(\overset{\max}{q_r}(CSS_i)\right) - ag\left(\overset{\min}{q_r}(CSS_i)\right)}, & \text{if } ag\left(\overset{\max}{q_r}(CSS_i)\right) \neq ag\left(\overset{\min}{q_r}(CSS_i)\right) \\ 1, & \text{otherwise} \end{cases} \tag{1}$$

$$\tilde{Q}_r(CMS) = \begin{cases} \frac{ag\left(\overset{\max}{q_r}(CSS_i)\right) - ag\left(q_r(CS_i^{j_i})\right)}{ag\left(\overset{\max}{q_r}(CSS_i)\right) - ag\left(\overset{\min}{q_r}(CSS_i)\right)}, & \text{if } ag\left(\overset{\max}{q_r}(CSS_i)\right) \neq ag\left(\overset{\min}{q_r}(CSS_i)\right) \\ 1, & \text{otherwise} \end{cases} \tag{2}$$

where one *CS* per *CSS* is selected to constitute a *CMS*, i.e., $CMS = \{CS_1^{j_1}, \cdots, CS_i^{j_i}, \cdots, CS_n^{j_n}\}$, $CS_i^{j_i}$ denotes the selected *CS* from CSS_i, $j_i \in [1, k]$ and $i = 1, 2, \cdots, n$; $q_r(CS_i^{j_i})$ denotes the value of *r*th QoS criteria for $CS_i^{j_i}$; $\overset{\max}{q_r}(CSS_i)$ and $\overset{\min}{q_r}(CSS_i)$ are, respectively, the maximum and minimum possible values of *r*thQoS criteria for CSs in CSS_i. The aggregation symbol (i.e., $ag = \{\Sigma, \prod, avg, \min \text{ and } \max\}$) can be used according to the characteristics of the QoS criteria and interconnection structures between component services, as shown in Table 1.

When MaOEAs are adopted to tackle SCOS, each QoS criterion corresponds to an objective function. Let $\mathbf{x} = \{x_1, \cdots, x_i, \cdots x_n\}$ be a *CMS*, where x_i represents the selected $CS_i^{j_i}$ for ST_i. The objective functions for the maximum quality criterion can be rewritten as follows in the context of minimization problems:

$$\begin{aligned} &\text{Minimize} \quad \mathbf{F}(\mathbf{x}) = \{f_1(\mathbf{x}), \cdots, f_r(\mathbf{x}), ..., f_m(\mathbf{x})\} \\ &\text{Subjectto}: \ \mathbf{x} \in \Omega \end{aligned} \tag{3}$$

where,

$$f_r(\mathbf{x}) = 1 - \tilde{Q}_r(CMS) \tag{4}$$

where $\mathbf{F}$: $\Omega \to \mathrm{R}^m$ consists of m objective functions (QoS criteria); R^m is the objective space; and $\mathbf{x}$ is the decision vector consisting of n variables. $\Omega = \prod_{i=1}^{n}[1, k_i] \in \mathrm{R}^n$ is called the decision space. Such a problem can be called many-objective SCOS if $m \geq 4$.

4. Proposed algorithm: MS-DABC

4.1. Framework of the proposed algorithm

The proposed Multi-population Self-adaption Differential ABC (MS-DABC) follows the same schema as the basic ABC, but involves crucial improvement strategies in many-objective contexts, such as food source updating, external archive (*EXA*)

Algorithm 1 General framework of MS-DABC.

```
Initialization()
while termination criterion is not fulfilled do
    g = g + 1;/*update generation counter*/
    Update_Parameters();
    Send_Employed_Bees();
    Send_Onlooker_Bees();
    Send_Scout_Bees();
    Update_Utility();
end
Output EXA;/*Return the Pareto front*/
```

maintenance, computational resource allocation among subpopulations, and potential search region detection in the onlooker bee phase. Specifically, several bee colonies evolve independently with different reproduction operators and exchange information periodically, and the population size for each colony can be adjusted based on the utility of the corresponding subpopulation, thereby resulting in a dynamical configuration of the computational efforts. The *EXA*, maintained by Pareto-dominance and decomposition-based selections in a steady state manner, is employed to measure the utility of the subpopulations. The main features of the proposed algorithm are summarized as follows:

(1) Reference vector guided mating restriction: in a many-objective space, arbitrarily selected solutions for the reproduction operation are likely to produce an offspring far away from its parents. This may cause an unacceptable distribution of solutions and lead to severe deterioration in the convergence of the algorithm. Thus, mating restriction is introduced so that mating parents can be chosen from a neighborhood, as much as possible.
(2) The adaptive synergy of multiple evolutionary operators: to exploit the merits of different operators, we combine multiple evolutionary operators by using dynamic configuration schema, inspired by the ideas in [38] and [21]. However, differing from the previous ensemble method concentrating on SOPs [38] or MOPs using decomposition based techniques (see [21]), in which the fitness improvement can be measured straightforwardly, here the selection of operators is based on their contributions to *EXA* members.
(3) Indicator based prominent search region detection: traditionally, in order to identify the potential search region in the onlooker bee phase, Pareto dominance and density information is employed to compute the fitness value of a given solution for problems involving two or three objectives (see [6,50]). However, due to the dominance resistance issue in many-objective space, such information fails to identify the difference between solutions and is naturally unsatisfactory in fitness assignment. To remedy this issue, a performance indicator is employed to achieve fitness assignment in prominent search region detection.
(4) Hierarchical elite-preserving of the *EXA*: a new elite-preserving strategy, which combines dominance and decomposition-based approaches, is used to balance the convergence and diversity of solutions in the *EXA*. The updating of the *EXA* is conducted based on nondominated sorting, the local niche-counts of the subregions, and scalarization functions in a hierarchical manner, where a worst solution in terms of dominance can survive in case it is associated with an isolated subregion. This is beneficial for achieving a balance between the convergence and diversity of the *EXA*.

The main procedure of MS-DABC is shown in Algorithm 1, which is constituted in four main parts: Initialization, Update Parameters, Bee Foraging Behavior (e.g., Send Employed Bees, Send Onlooker Bees and Send Scout Bees) and Update Utility. Note that the archive management strategy is embedded in the subparts of Bee Foraging Behavior in a steady state manner, computational resource allocation is achieved in Update Utility, and the parameter adaptation of reproduction operators is realized in Update Parameters. To deal with MaOPs with badly scaled PFs, the objective vectors of the solutions are normalized in each generation by ideal and nadir points, as suggested in [39]. In the following sections, the key components of MS-DABC are given after the introduction of some preliminary knowledge of ABC.

4.2. Basic ABC algorithm

ABC [15] is inspired by the food foraging behavior of honeybees. The optimization procedure of ABC starts by the initialization similar with other metaheuristics. Then the whole bee colony is divided into three subgroups, i.e., the employed bees, onlookers and scouts, to perform different tasks until the termination condition is satisfied. Employed bees try to explore new food sources by learning from neighbors and then share the nectar amount of the food sources with onlooker bees. Given a food source $\mathbf{x}_i$, the employed bee i tries to find out a temporary position $\mathbf{v}_i$ by the following equation:

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) \tag{5}$$

where x_{ij} and v_{ij} represent the jth variable of $\mathbf{x}_i$ and $\mathbf{v}_i$, respectively; φ_{ij} is randomly selected in range $[-1, 1]$; i, k and j are random integers with $i, k \in [1, N]$ $(i \neq k)$, and $j \in [1, n]$. N and n are the number of food sources and dimension of search space

Algorithm 2 Initialization().

/* *The setting of control parameters* */
Set generation counter $g=0$, utility update period ng, the abandonment criteria *limit*, set the control parameters of reproduction parameters in subpopulations $\mu CR_k = 0.5$, $\mu F_k = 0.5$, $\pi^k = 0$, $S_{CR,k} = \phi$, $S_{CR,k} = \phi$, $S_{F,k} = \phi$, and $\Delta F_{es_k} = 0$ for $k = 1, 2, 3, 4$;
/* *The initialization of food sources* */
Generate an initial population $pop = \{X^1, \cdots X^N\}$ by random sampling from Ω;
Randomly partition *pop* into $subpop_k (k = 1, 2, 3, 4)$ with respect to their sizes;
Initialize ideal point $Z^* = (Z_1^*, \cdots Z_m^*)$ by setting $z_i^* = \min\{f_i(x^1), \cdots f_i(x^N)\}$:
$EXA \leftarrow pop$;
/* *Generation of reference* (*weight*) *vectors* */
Generate a set of reference vectors $\Lambda = \{\lambda^1, \ldots \lambda^N\}$;
Perform the assignment of neighborhood $E(i) = \{i_1, \cdots, i_T\}$, where $\lambda^{i_1}, \cdots \lambda^{i_T}$ are the closest vectors to λ^i based on the Euclidean distance;

respectively. For a minimization problem, the fitness value for $\mathbf{v}_i$ can be calculated by:

$$fit_i = \begin{cases} 1/(1+f_i) & \text{if } f_i \geq 0 \\ 1 + abs(f_i) & \text{if } f_i < 0 \end{cases} \tag{6}$$

According to the quality (nectar amount) of the food source shared by the employed bees, an onlooker probabilistically chooses to visit a food source by using equation below:

$$p_i = fit_i \Big/ \sum_{k=1}^{N} fit_k \tag{7}$$

where fit_i represents the fitness of the ith solution, $i \in \{1,2,\ldots, N\}$. Then a new food source is generated by using Eq. (5) and the better one is memorized for further exploitation in the next iterations.

If the quality of a food source cannot be improved by a predetermined number of times named *limit*, the bee of a corresponding food source becomes a scout that explores a new food source randomly.

4.3. Initialization

The initialization procedure of MS-DABC is provided in Algorithm 2, which contains three main aspects: the setting of control parameters, the initialization of food sources, and the generation of reference (weight) vectors. Parameters involved in the algorithm are introduced in the following sections. The initialization of food sources is the same as the original ABC aforementioned. To alleviate the blindness of parents mating for offspring production, the bee colony uses the neighborhood concept of MOEA/D [45] for mating restriction, each food source is associated with a subproblem, and each subproblem is predefined by a unique weight vector, which also specifies a search subregion in the objective space.

To be specific, a set of evenly spread weight vectors $\Lambda = \{\lambda^1, \ldots, \lambda^N\}$ is generated by using a simplex-lattice design method suggested in [45]. In the case of objective number $m > 5$, the two-layer weight vector generation method proposed in [20] is adopted for better uniformity. Each weight vector $\lambda^i = (\lambda_1^i, \lambda_2^i, \ldots, \lambda_m^i)$, specifies a unique search region Ψ^i, which is defined as:

$$\Psi^i = \{F(x) | \theta^i(x) \leq \theta^j(x), \forall j \in \{1, \cdots, N\}\} \tag{8}$$

$$\theta^i(x) = \left\langle \lambda^i, F(x) - z^* \right\rangle = \arccos\left(\frac{\lambda^i \times (F(x) - z^*)}{\left\| \lambda^i \right\| \times \| F(x) - z^* \|} \right) \tag{9}$$

where $i \in \{1, \ldots, N\}$, $x \in \Omega$, and $\theta^i(x)$ is the acute angle between λ^i and $F(x)$-z^*, z^* is the ideal point. Thereafter, for a solution x, the index of its associated subregion Ψ^k is defined as:

$$k = \arg\min_{i \in \{1, \cdots, N\}} \left\langle F(x) - z^*, \lambda^i \right\rangle \tag{10}$$

By introducing the neighbor concept of MOEA/D, each food source is associated with a weight vector, which has been assigned with a neighborhood based on the Euclidean distance. Thereafter, each food source can be optimized by using food sources chosen from its neighbor subregions.

4.4. Food source update

The proposed algorithm maintains two types of populations, the current population *pop* and the *EXA*. To update *pop*, the scalar optimization subproblems of MOEA/D are employed to estimate the quality of the solutions. For each weight vector in MOEA/D, the resulting problem is called a subproblem, which can evaluate the fitness value of a solution. Three popular

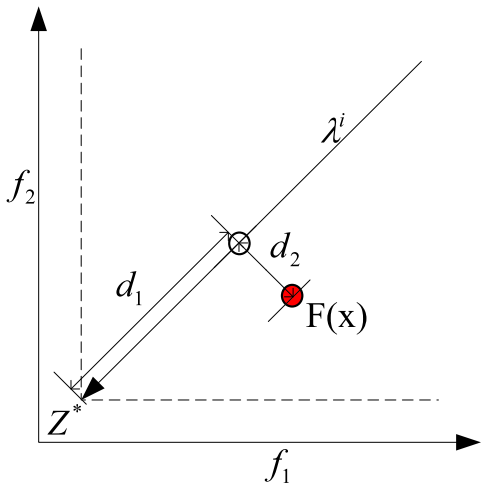

Fig. 3. Illustration of PBI.

aggregation approaches, the weighted sum, weighted Tchebycheff and penalty-based boundary intersection (PBI), can be used to form subproblems. The PBI approach is adopted in this paper, due to its advantages in handling many-objective problems. The PBI approach is defined as:

$$\text{Minimize } g^{pbi}\left(x\middle|\lambda^i, z^*\right) = d_1 + \theta d_2 \tag{11}$$

$$\text{where } d_1 = \frac{\left\|(F(x) - z^*)^T \lambda^i\right\|}{\left\|\lambda^i\right\|} \tag{12}$$

$$\text{and } d_2 = \left\| F(x) - \left(z^* - d_1 \frac{\lambda^i}{\left\|\lambda^i\right\|} \right) \right\| \tag{13}$$

where θ is a user-defined penalty factor; d_1 and d_2 are the length of the projection of vector (F(x)-z^*) on the weight vector λ^i and the perpendicular distance from F(x) to λ^i, respectively. $z^* = (z_1^*, \cdots, z_m^*)$ is the ideal point and its ith component can be computed by:

$$z_i^* = \min_{x \in \Omega} f_i(x) \tag{14}$$

Fig. 3 presents a simple illustration of the PBI approach.

The updating procedure of *pop* is based on local mating and local replacement. Local mating considers selecting parents with mating restriction (e.g., from neighboring subregions) to produce offspring while local replacement means each child solution replaces, at most, a certain number of solutions in its neighboring subregions. Specifically, given a subproblem i, an offspring solution x_c will be generated by applying reproduction operators on some selected parents from the neighboring subregions. Then, x_c is used to replace at most n_r solutions in *pop* if it performs better than them in terms of their respective subproblems' aggregation values.

4.5. Following probability of onlooker bees

Traditionally, the fitness involved in Eq. (7) for MOEAs is evaluated based on Pareto dominance relationship and the density information (such as crowding distance [6] in NSGA-II and kth nearest distance [50] in SPEA2), and the former is always preferable to the latter. Such mechanism provides more opportunity for solutions being the first Pareto front but possibly with poor diversity, so it will bias the search toward a small segment of PF. On the other hand, Pareto dominance fails to distinguish solutions in the high-dimensional objective space [5], and fitness assignment based purely on crowding distance may cause loss of selection pressure for solutions to converge toward the PF, and the search will be largely directionless. Inspired by the promising performance of indicator-based MOEAs in addressing MaOPs, indicator-based fitness assignment is employed to guide the food source selection for onlookers.

In principle, any performance indicator (e.g., HV [1] and epsilon metric $I_{\varepsilon+}$ [37]) can be applied in fitness evaluation. In this study, we use the epsilon metric $I_{\varepsilon+}$, due to the high computational cost of HV. To be specific, the fitness of a food source x is estimated based on epsilon metric:

$$fit(x) = \sum_{x' \in P \setminus x} -e^{-I_{\varepsilon+}(x', x)/k} \tag{15}$$

$$\text{subject to } I_{\varepsilon+}(\mathrm{x}', \mathrm{x}) = \min\left\{\varepsilon | f_i(\mathrm{x}') - \varepsilon \le f_i(\mathrm{x}), i \in \{1, \cdots, m\}\right\} \tag{16}$$

where P represents the current population, parameter k is a scaling factor that is set as 0.05 in this study. Eq. (15) approximates the HV contributions of a solution regarding the optimization goal by estimating the sum of ε needed to cover x by the remaining members of the population P. Eq. (16) gives the minimum distance in which solution x' can be translated in one possible dimension of the objective space such that x is weakly dominated. In essence, $fit(\mathrm{x})$ is adopted to estimate the "loss in quality" of the whole solution set if x is removed away from P. Note that the higher the value of fitness, the better the quality of the food source. To avoid a solution with the extremely largest fitness always being selected, hence causing the population prematurely to converge into a local optimum, instead of using the raw fitness value, the ranking of population members is used to assign the following probability of onlookers. We can sort the population members according to fitness in descending order. The ranks of individuals are then assigned as follows:

$$Rank_i = N - i, \quad i = 1, 2, \cdots, N \tag{17}$$

where N is the population size. Then the following probability p_i of the ith food source is defined as:

$$p_i = Rank_i / N, \quad i = 1, 2, \cdots, N \tag{18}$$

4.6. Synergy of adaptive reproduction operators with complementary effect

In our proposal, MS-DABC partitions the whole population into four subgroups evolving with a distinct reproduction operator. The adaption of control parameters for these operators is discussed in detail in Section 4.7. In addition, subpopulations' sizes are dynamically adjusted based on their operators' performance, with related details explained in Section 4.8.

Regarding the operator selection, several recommendations have been provided in [3,27,29,43] and their settings are determined by the characteristics of the problem. The original DE adopts the "rand/1" mutation operation, which exhibits strong exploration ability but poor convergence performance. For the "rand/2" mutation operation, it leads to more perturbation than "rand/1", as more target vectors are selected. The "current-to-pbest/1" mutation operation was introduced to improve search robustness by exploiting the historical record of success [43]. The "current-to-rand/1" mutation operation is very suitable for handling rotated problems due to the rotation-like perturbation [29]. Mutation operations like "best/1"and "best/2" introduce the global best guided term in the update equation, which may cause premature results when solving problems with complex fitness landscapes. Therefore, these mutation operations have not been employed in MS-DABC.

Based on the above consideration, four mutation operations with complementary effects, i.e., "rand/1", "rand/2", "rand-to-best/2" and "current-to-pbest/1"are adopted in the four subgroups respectively. The combination of those mutation operations with binomial crossover forms a number of evolutionary operators. Given a solution $\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{in})$, the new candidate solution $\mathbf{v}_i = (v_{i1}, v_{i2}, \ldots, v_{in})$ is produced as follows:

"rand/1/bin"

$$v_{i,j} = \begin{cases} x_{r_1,j} + F(x_{r_2,j} - x_{r_3,j}), & \text{if } (rand(0,1) \le CR) \text{ or } (j = j_{rand}) \\ x_{i,j}, & \text{otherwise} \end{cases} \tag{19}$$

"rand/2/bin"

$$v_{i,j} = \begin{cases} x_{r_1,j} + F(x_{r_2,j} - x_{r_3,j}) + F(x_{r_4,j} - x_{r_5,j}), & \text{if } (rand(0,1) \le CR) \text{ or } (j = j_{rand}) \\ x_{i,j}, & \text{otherwise} \end{cases} \tag{20}$$

"rand-to-best/2/bin"

$$v_{i,j} = \begin{cases} x_{i,j} + F(x_{best,j} - x_{i,j}) + F(x_{r_1,j} - x_{r_2,j}) + F(x_{r_3,j} - x_{r_4,j}), & \text{if } (rand(0,1) \le CR) \text{ or } (j = j_{rand}) \\ x_{i,j}, & \text{otherwise} \end{cases} \tag{21}$$

"current-to-pbest/1/bin"

$$v_{i,j} = \begin{cases} x_{i,j} + F(x_{pbest,i,j} - x_{i,j} + x_{r_1,j} - \tilde{x}_{r_2,j}), & \text{if } (rand(0,1) \le CR) \text{ or } (j = j_{rand}) \\ x_{i,j}, & \text{otherwise} \end{cases} \tag{22}$$

where indices r_1, r_2, r_3, r_4, and r_5 are random mutually exclusive integers within the range [1, N]; $j \in \{1, 2, \ldots, n\}$; F is the scaling factor; CR is crossover rate; Let op_1, op_2, op_3, op_4 denote operators in Eqs. (19)–(22), respectively; $x_{best,j}$ in op_3 indicates the jth element of $\mathbf{x}_{best}$, which is a randomly selected global best solution from the *EXA*; $x_{pbest,i,j}$ in op_4 denotes the jth element of $\mathbf{x}_{pbest,i}$, which represents the personal best of $\mathbf{x}_i$, i.e., a nondominated archive of personal best; and $\tilde{x}_{r_2,j}$ in op_4 represents the jth element of $\tilde{\mathbf{x}}_{r_2}$, which is picked up from the union set of evolving population and archive set. Note that the archive for op_4 provides information about the search direction and is devoted to strengthen the diversity, as pointed out by [43].

4.7. Parameters adaptation for reproduction operators

Various conflicting conclusions have been drawn with regard to the manual parameter tuning of scaling factor F and crossover rate CR [27], which were performed in different problem contexts and hence lack generality. In fact, different parameter settings may be required at different stages of evolution so as to achieve the best performance. In our empirical study of self-adaptive parameter strategies proposed in [3,29,43], the one in [43] was verified to be the most suitable. Thus, it is chosen to be the parameter adaption approach in MS-DABC.

At generation g, let $CR_{i,k}$ be the crossover rate of kth reproduction operator used by individual $\mathbf{x}_i$, sampled through the follow equation:

$$CR_{i,k} = \text{randn}_{i,k}(\mu CR_k,\ 0.1) \tag{23}$$

where $CR_{i,k}$ is confined in the range [0, 1], μCR_k represents the average value of $CR_{i,k}$ and is set to 0.5 in initialization. Assumed that $S_{CR,k}$ be the set of successful $CR_{i,k}$, then μCR_k, at generation g, is evaluated by:

$$\mu CR_k = (1-c)\cdot \mu CR_k + c\cdot \text{mean}_A(S_{CR,k}) \tag{24}$$

where c is a constant parameter within [0.05, 0.2], $\text{mean}_A(.)$ denotes the arithmetic mean function.

Similarly, the scaling factor of the kth reproduction operator used by $\mathbf{x}_i$, i.e., $F_{i,k}$ is updated according to Cauchy distribution:

$$F_{i,k} = \text{randc}_{i,k}(\mu F_k,\ 0.1) \tag{25}$$

where μF_k denotes the location parameter, and 0.1 is the scale parameter. $F_{i,k}\in[0, 1]$.

Let $S_{F,k}$ be the set of successful $CR_{i,k}$ at generation g, and μF_k is set to 0.5 initially, so at generation g, the location parameter μF_k is updated as:

$$\mu F_k = (1-c)\cdot \mu F_k + c\cdot \text{mean}_L(S_{F,k}) \tag{26}$$

where $\text{mean}_L(.)$ denotes the Lehmer function:

$$\text{mean}_L = \frac{\sum_{F\in S_F} F^2}{\sum_{F\in S_F} F} \tag{27}$$

Then, the scaling factor and crossover rate are adapted and sampled from gradually adaptive probability distributions.

4.8. Subpopulation size adaption

From the perspective of resource conservation, computational efforts should be distributed to the subpopulations based on their utilities, which are obtained through the analysis of search data from the previous evolutionary process. In this study, we define and calculate a utility π^k for each subpopulation k. During every ng generation, the utility π^k will be updated and the sizes of subpopulations adjusted correspondingly.

In detail, let pop be the whole population, and $subpop_1$, $subpop_2$, $subpop_3$ and $subpop_4$ be the subpopulations of pop. Clearly, we have:

$$pop = \bigcup\nolimits_{k=1,\dots K} subpop_k \tag{28}$$

Let the pop size be N, and the size of $subpop_k$ be N_k, γ_k is the ratio between $subpop_k$ and pop, then:

$$N_k = \gamma_k \cdot N, \quad k = 1, 2, ..., K \tag{29}$$

where $\gamma_k \in [0, 1]$, $\sum_{k=1}^{K}\gamma_k = 1$, and K is the total number of subpopulations.

The EXA, which is detailed in Section 4.9, is employed to reserve the elite solutions found and update right after the generation of a new candidate. A flag is then marked to each member of EXA to show it is obtained by which subpopulation. If a solution is found by more than one subgroup, the flag contains all indexes of these subgroups. Let b_g^k be the number of EXA members provided by $subpop_k$ in the gth generation. At the generation G, the utility π^k is updated as:

$$\pi^k = \sum_{g=G-ng}^{G-1} b_g^k / \Delta Fes_k, \quad k = 1, 2, ..., K \tag{30}$$

where G is the current generation, $\sum_{g=G-ng}^{G-1} b_g^k$ is the total number of EXA members found by $subpop_k$ within the last ng generations, and ΔFes_k is the consumed function evaluations by $subpop_k$ within the last ng generations. Then, the ratio between $subpop_k$ and pop, i.e., γ_k, can be evaluated as follows:

$$\gamma_k = \pi^k / \sum\nolimits_{k=1}^{K}\pi^k, \quad k = 1, 2, ..., K \tag{31}$$

and the size of $subpop_k$ can be evaluated:

$$N_k = \gamma_k \cdot N = \left(\pi^k \Big/ \sum\nolimits_{k=1}^{K}\pi^k\right)\cdot N, \quad k = 1, 2, \dots, K \tag{32}$$

A minimum size (i.e., 5) is set for all subpopulations to guarantee that each of them has a chance to evolve. Clearly, the higher the utility for a subgroup, the more individuals are assigned to it, which means that better performed subpopulations are allocated with more computational resources. With the algorithm proceeding, the utility recalculation and population partitioning operation are executed periodically. At the beginning, the subpopulations are set to the same size. If the current generation number G is a multiple of ng, then compute π^k for each subpopulation k during the last ng generations, and update the sizes of all subpopulations. In addition, to maintain the diversity of population and facilitate information exchange among subpopulations, the members of the subpopulations are randomly picked from the whole population once the regrouping operation is triggered.

4.9. Elite-preserving strategy

To alleviate the difficulties resulting from different methods, we integrate the merits of Pareto dominance, niche-preservation and decomposition-based approaches to guide the elite-preserving process in a hierarchical manner. In particular, the balance between the proximity and distribution of *EXA* is achieved by two different modules, i.e., Pareto-based Ranking (PR), and Niche & Decomposition based Selection (NDS), separately. PR is only employed as a sorting operator to rank *EXA* members into different Non-domination Levels (NDLs), while a well-designed NDS, in which the density indicator (i.e., niche count) is combined with the decomposition-based technique, can be used to prune the inferior solutions of *EXA* in a more fine-grained manner.

It should be pointed out that, instead of performing *EXA* updating after all members of the whole offspring are generated, *EXA* is updated right after the generation of a new candidate in a steady-state manner, so that the elite information can be utilized as soon as possible. More specifically, at each generation for each solution $\mathbf{x}_i$, an offspring $\mathbf{x}_c$ is generated by performing a reproduction operator on several solutions selected from the neighbor subregions of $\mathbf{x}_i$. Then, $\mathbf{x}_c$ is combined with *EXA* to form a intermediate *EXA'* . Afterwards, one of inferior solutions from *EXA'* is identified and eliminated. The method in [19] is employed for the updating of NDLs so that the computational cost is reduced.

As mentioned earlier, identification of the worst solution x' from *EXA'* is made in a hierarchical manner. At the beginning, the intermediate *EXA'*, a combination of *EXA* and $\mathbf{X}_c$, is divided into several NDLs, i.e., $F_1, \cdots, F_l$. Afterwards, we identify the most crowded subregion associated with solutions in F_l, denoted as $\Psi^{crowd}_{x\in F_l}(x)$, where the crowded degree of a subregion is determined by the number of solutions (niche count) associated with it. Then, the following two scenarios are considered according to the number of solutions in $\Psi^{crowd}_{x\in F_l}(x)$, i.e., $|\Psi^{crowd}_{x\in F_l}(x)|$:

***Case* 1**: $|\Psi^{crowd}_{x\in F_l}(x)|=1$, i.e., only one solution exists in the most crowed subregion associated with those solutions in F_l, (if $|F_l|>1$, then each member of F_l is located in an isolated area) which means that $\Psi^{crowd}_{x\in F_l}(x)$ is likely to be an unexploited subregion. In this case, the solution in $\Psi^{crowd}_{x\in F_l}(x)$ should be reserved for the next round of exploitation. Instead, we identify the most crowded subregion associated with solutions in F_{l-1}, so on and so forth. Once$|\Psi^{crowd}_{x\in F_{l-k}}(x)|>1(k=0,\ldots,l-1)$, we find the worst solution x' from $\Psi^{crowd}_{x\in F_{l-k}}(x)$ in terms of the largest aggregation function value of the subproblem, and x' is eliminated from *EXA'* . In the extreme case, $|\Psi^{crowd}_{x\in F_{l-k}}(x)|=1(k=l-1)$, which means each member of *EXA'* is associated with an isolated subregion. In this case, solution x' from F_l with the worst aggregation function value of the subproblem is eliminated from *EXA'* . An example of this is illustrated in Fig. 4(a), where the *EXA'* is divided into 4 NDLs, i.e., $F_1=\{x^1,x^3\}$, $F_2=\{x^5,x^6\}$, $F_3=\{x^4\}$, and $F_4=\{x^2\}$, the most crowed subregion associated with these solutions in F_l is Ψ^4, which has only one member x^2, thus Ψ^4 is likely to be an unexploited subregion, and x^2should be reserved. Next, the same goes for x^4 in F_3, which should be reserved. Afterwards, subregions associated with these solutions in F_2, i.e. Ψ^2 and Ψ^5, are investigated, and apparently the more crowded one is Ψ^5, which has two members x^3 and x^6. Finally, x^6 is eliminated since it has the largest aggregation function value in the corresponding subregion.

***Case* 2**: $|\Psi^{crowd}_{x\in F_l}(x)|>1$, i.e., multiple solutions exist in $\Psi^{crowd}_{x\in F_l}(x)$. The worst solution from which, identified by the highest aggregation function value of corresponding subproblem, is eliminated from *EXA'* . Such an example is illustrated in Fig. 4(b), where the subregion associated with solutions in F_l is Ψ^4. Apparently $|\Psi^4|>1$, so x^2 is removed from *EXA'* because it has the worst aggregation function value and there is a better solution x^6 for the same subproblem in terms of convergence.

It should be pointed out that, during the identification of the most crowded subregion, there may be multiple subregions having the same niche count. In this case, the one with the largest sum of aggregation function values is selected:

$$c=\arg\max_{j\in S}\sum_{x\in\Omega^j} g^j(x) \tag{33}$$

where S denotes the indices set of subregions with the same crowded degree, Ω^j represents the subregion of the corresponding subproblem j, and $g^j(x)$ indicates the aggregation function of the subproblem j for the current solution x. The procedure of the *EXA* update is presented in Algorithm 3, where the recurrence method of worst member identification is given in Algorithm 4.

Differing from NSGA-III, where Pareto dominance is considered as a major selection criteria, Pareto based ranking in our proposed approach is only employed as a sorting operator instead of selection criteria, and the selection is based on

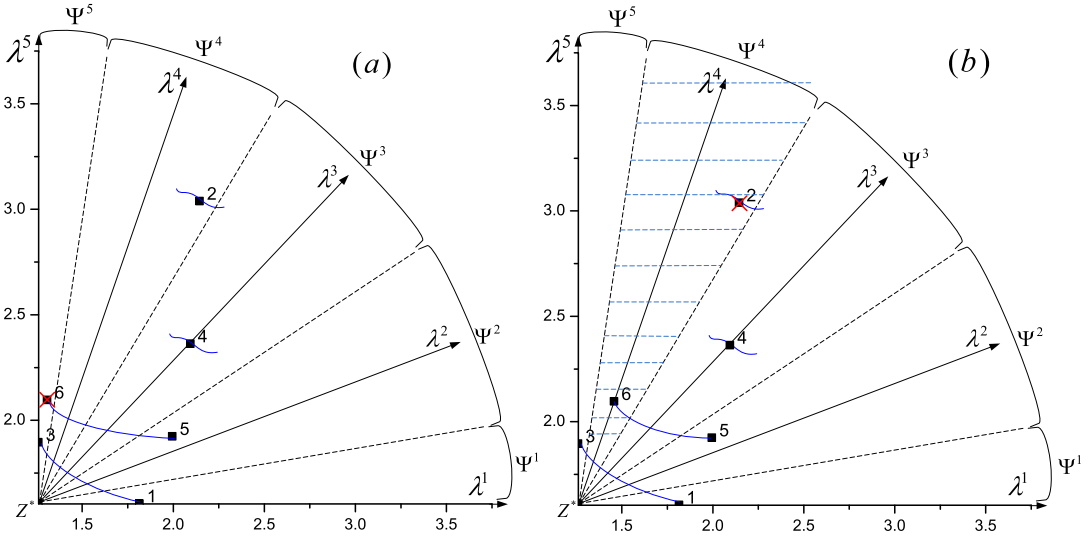

Fig. 4. Examples of the worst member elimination in *EXA*.

Algorithm 3 UPDATEEXA (*EXA*, x_c).

Input: external archive EXA; offspring solution x_c
Output: external archive *EXA*
$EXA' \leftarrow EXA \cup x_c$
$NDLs = \{F_1, \cdots, F_{end}\}$: %∗ Use the method in [19] to update the nondomination level structure of EXA^*/
$l = |NDLs|$
if $|\Psi^{crowd}_{x \in F_l}(x)| == 1$ **then**
  if $l! = 1$ **then**
    $l \leftarrow l - 1$
    $x' \leftarrow$ ID_worst($NDLs, EXA', l$)
  else // $l == 1$
    $x' \leftarrow$ identify the worst member of F_{end} in terms of *PBI*
  end
else
  Identify the most crowded subregion Ω^d
  associated with those solutions in F_l
  $x' \leftarrow \text{argmax} g^d(x)$, $x \in \Omega^d$
end
$EXA \leftarrow EXA' \backslash x'$

the density indicator and aggregation functions sequentially. In particular, the density indicator, which is based on niche counts, concerns the diversity issue; while the selection operation, determined by aggregation function values, concerns the convergence issue. In other words, it is a diversity-first and convergence-second method. For the case in Fig. 4(a), due to the dominance priority strategy of NSGA-III, solution x^2 is removed without reservation since it is located in the last NDL. However, in our proposed approach, solution x^2 is reserved as it is associated with an isolation subregion. Instead, solution x^6 is removed due to its worst aggregation function value in the most crowded subregion. In this study, a selection priority is provided to the member F_l in case it is located in an isolated subregion. On the contrary, since solutions in the former NDL have higher priority in NSGA-III, x^2 is eliminated and the population diversity is not well-maintained.

4.10. The implementation procedure of MS-DABC for SCOS problems

Based the above analysis, the detailed implementation of MS-DABC for solving MaOPs is presented in Algorithm 5, where the algorithm starts by initializing the bee colony and setting the parameters, then enters the evolution loop until the maximum number of generations, *maxG*, is achieved. At first, the colony is randomly divided into several subgroups with equal sizes, which are independently evolved by employing different evolutionary operators. The *EXA*, which is maintained by a fine-grained selection mechanism, is used to guide the search process. Based on the contributions of the evolutionary operators for the *EXA* members, the sizes of the subgroups are adjusted periodically. Parameter updating for the evolutionary

Algorithm 4 ID_worst ($NDLs, EXA, l$).

Input: external archive EXA; nondomination level structure of EXA: $NDLs$; current level :l
Output: the worst member of EXA: x′
if $|\Psi^{crowd}_{x\in F_l}(x)| == 1$ **then**
  if $l! = 1$ **then**
    $l \leftarrow l - 1$
    x′ ← ID_worst($NDLs, EXA, l$)
  else// $l == 1$
    x′ ← identify the worst member of F_{end} in terms of PBI
  end
else
  Identify the most crowded subregion Ω^d
  associated with those solutions in F_l
  x′ ← argmaxg(x), x ∈ Ω^d
end

operators is given in lines 5–9, regrouping of subpopulation is given in lines 32–39; the food source position update as well as *EXA* maintaining strategy are given in Algorithm 6; and other key components are elaborated before.

5. Experiments on benchmark MOPs and MaOPs

A series of experiments are carried out with the aim of testing the performance of the proposed algorithm in solving MOPs and MaOPs. In Section 5.1, experimental settings are briefly described at first, and in Section 5.2, we compare the proposed algorithm with several state-of-the-art MOEAs having a competitive performance. Then, the scalability of proposed algorithm for tackling many-objective problems is investigated in Section 5.3. Next, in Section 5.4, the aim is to evaluate the effectiveness of improvement strategies for MS-DABC. Finally, experiments in Section 5.5 are performed to investigate the sensitivity of parameters on the proposed algorithm.

5.1. Experimental settings

(1) *Test problems*: Several types of benchmark MOPs with various characteristics and different difficulties are chosen in our experimental studies, including: 1) the tri-objective DTLZ family of scalable instances: DTLZ1-DTLZ7; and 2) the bi-objective WFG toolkit problems: WFG1-WFG9. The above test instances have various features such as nonlinear or degenerate geometry, variable linkage, and multi-modal, which pose a great challenge to MOEAs. More details of the test MOPs are described in [1,2].
(2) *Performance metrics*: In order to evaluate the performance of MOEAs on the test MOPs, both inverted generational distance (IGD) and hyper-volume (HV) metrics can be used to measure the convergence and diversity of the obtained solutions set. However, the computational complexity of HV is too high for PFs with many objectives. Therefore, only IGD is employed as the performance metric in our experiments. The detailed calculation of IGD metric has been elaborated in [12,23]. A low value of IGD means that the obtained solution set is very close to the true PF. In addition, indicator SPREAD ($\triangle$) is employed to measure the uniformity of the solution set found, as suggested in [39].
(3) *Parameters setting*: In our study, the proposed algorithm is compared with several state-of-the-art MOEAs. Parameters settings of the MOEAs are adopted as recommended in the literature, as summarized in Table 2, where n is the variable dimension, p_m and p_c are the mutation and crossover probabilities respectively, η_m and η_c are the distribution indexes of crossover and mutation operators respectively, T is the neighborhood size of weight vectors, n_r is the maximum number of solutions replaced by each new solution, and δ is the probability of selecting parent solutions from neighborhood. The population size N and the maximum function evaluations (Max_FEs) are adjusted according to the complexity of MOPs. N is set as 200, and $Max\_FEs = 50{,}000$. To allow a fair comparison, the settings of T, δ, and n_r for MS-DABC are the same as other algorithms, since the control parameters involved in the evolutionary operators mentioned in Section 4.6 can be dynamically configured, and only four parameters need to be set, i.e., N, $maxG$, $limit$, and ng. Among them, N is a common parameters and is set the same as other peer algorithms, $maxG$ is decided by Max_FEs and N, and $limit$ is set as $20n$ as suggested by [14]. Thus, only ng needs to be tuned and its value is set as 30 based on the sensitivity analysis provided in Section 5.5.

For fair comparison, all the algorithms take the same Max_FEs for the same instance. To get statistically sound conclusions, the compared algorithms are executed 30 times for each test instance, and the mean and the standard deviations (std) of the metric values are recorded for comparison. A non-parametric test (i.e., Wilcoxon's rank sum test) at a 0.05 significance level is further performed to assess the statistical significance of the results obtained by the compared MOEAs.

Algorithm 5 MS-DABC.

Algorithm 5: MS-DABC

Initialization() /*detailed in Sec.4.3*/
while $g<maxG$
  $g=g+1$;
  /*Parameter update for Reproduction operators, detailed in Sec.4.7*/
  for k=1:4
    Update μCR_k by Eq.(24), and μF_k by Eqs. (26) and (27) respectively in case of $S_{CR,k}\neq\phi$ and $S_{F,k}\neq\phi$;
    Sample $CR_{i,k}$ and $F_{i,k}$ for each individual $\mathbf{x}_i$ in $subpop_k$ by Eq.(23) and Eq.(25)
    Set $S_{CR,k}=\phi$ and $S_{F,k}=\phi$;
  end
  /*Employed bee phase*/
  for i=1:N
    Perform **Algorithm 6** to update solutions and *EXA*; /*detailed in Sec.4.4 and Sec.4.9*/
  end
  /*onlooker bee phase*/
  Calculate probabilities $p(i)$ for onlookers by Eq.(15)~Eq.(18)/*following probability, detailed in Sec.4.5*/
  $i\leftarrow 1; s\leftarrow 0$;
  while $s<N$ //Onlooker bee phase: send onlookers to food sources
    $r\leftarrow rand(0,1)$;
    if $r<p(i)$ then //Stochastic sampling;
      $s\leftarrow s+1$;
      Perform **Algorithm 6** to update solutions and *EXA*; /*detailed in Sec.4.4 and Sec.4.9*/
    end
    $i\leftarrow (i+1)$ mod (N);
  end
  /*scout bee phase*/
  Determine a food source k with maximum *trial* value;
  if $trial_k>=limit$
    $\mathbf{x}_k\leftarrow$ a random solution;
    Evaluate the new food source $\mathbf{x}_k$; $trial_k=0$;
  end
  /*regroup subpopulations, detailed in Sec.4.8 */
  if mod(g, ng)==0
    for k=1:4
      Calculate the utilities for $subpop_k$ by Eq.(30);
      Update the size of $subpop_k$ according to Eq.(31) and Eq.(32);
      $\pi^k=0$; $\triangle Fes_k=0$;
    end
    Randomly regroup the population with respect to their updated sizes;
  end
end
Output *EXA* //return the Pareto optimal Front

Algorithm 6 Food source update.

Algorithm 6: Food source update

Input: Parent population pop; Reference vectors Λ; Neighborhood set of reference vectors E; External archive EXA; The total number of function evaluations consumed by $subpop_k$: ΔFes_k; Successful parameter set $S_{CR,k}$ and $S_{F,k}$; abandonment criteria $trial_i$;
Output: pop; EXA; ΔFes_k; $S_{CR,k}$; $S_{F,k}$; $trial_i$;
for $k = 1:4$ /* Food source updating for $subpop_k$ (k=1,2,3,4)*/
  if $x_i \in subpop_k$
    $Pmat \leftarrow \text{MatingResriction}(E(i), pop)$
    $x^c \leftarrow op_k(i, Pmat, pop)$ %Generate an offspring x_c by perform op_k on mating parents according to Eq.(19)~(22)
    Update the current reference point z^*
    $t = 0$
    while $t < n_r \parallel Pmat != \phi$ **do**
      Random pick a indice j from $Pmat$
      if $g(x_j|\lambda^j, z^*) - g(x_c|\lambda^j, z^*) > 0$
        $x_j \leftarrow x_c$
        $Pmat \leftarrow Pmat \setminus j$
        $trial_j = 0$;
        $S_{CR,k} \leftarrow C_{Ri,k}$; $S_{F,k} \leftarrow F_{i,k}$
      else
        $trial_j = trial_j + 1$
      end
      $t++$
    end
    $EXA \leftarrow \text{UpdateEXA}(EXA, x_c)$ //detailed in **Algorithm3**
    $\Delta Fes_k = \Delta Fes_k + 1$
  end
end

Table 2
Parameter settings for the compared algorithms.

Algorithms	Parameter settings
NSGA-II	$p_c = 0.9$, $p_m = 1/n$, $\eta_c = 20$, $\eta_m = 20$
SMPSO	$C_1 \in [1.5, 2.5]$, $C_2 \in [1.5, 2.5]$, $p_m = 1/n$, $\eta_m = 20$
SMS-EMOA	$p_c = 0.9$, $p_m = 1/n$, $\eta_c = 20$, $\eta_m = 20$
HypE	$p_c = 0.9$, $p_m = 1/n$, $\eta_c = 20$, $\eta_m = 20$
MOEA/D-DE	$p_m = 1/n$, $\eta_m = 20$, $CR = 1$, $F = 0.5$, $\delta = 0.9$, $T = 20$, $n_r = 2$
MOEA/D-DRA	$p_m = 1/n$, $\eta_m = 20$, $CR = 1$, $F = 0.5$, $\delta = 0.9$, $T = 0.1*N$, $n_r = 0.01*N$
ENS-MOEA/D	$N = 100$, $CR = 1$, $F = 0.5$, $LP = 50$, $\varepsilon = 0.01$
MOEA/D-FRRMAB	$CR = 1$, $F = 0.5$, $p_m = 1/n$, $\eta_m = 20$, $T = 20$, $\delta = 0.9$, $n_r = 2$, $C = 5.0$, $W = 0.5*N$, $D = 1.0$
MS-DABC	$limit = 20n$, $ng = 30$, $\mu CR = 0.5$, $\mu F = 0.5$, $T = 20$, $\delta = 0.9$, $n_r = 2$

5.2. Comparison of MS-DABC with state-of-the-art MOEAs

The MS-DABC is compared with state-of-the-art MOEAs including NSGA-II[1] [6], SMPSO[2] [28], SMS-EMOA[2] [2], HypE[3] [1], MOEA/D-DE[4] [18], MOEA/D-DRA[4] [44], ENS-MOEA/D[5] [47], and MOEA/D-FRRMAB[6] [21]. Particularly, NSGA-II and SMPSO are two well-known Pareto- based MOEAs, HypE and SMS-EMOA are two representative indicator-based MOEAs, and the rest are decomposition-based ones. The reasons we select those variants of MOEA/D for peer comparison are explained as follows. First, MOEA/D-DE uses extra measures for maintaining population diversity, which is very promising in tackling MOPs with complicated PS shapes. Second, MOEA/D-DRA employs utility metrics for subproblems to decide on promising search regions, which can achieve dynamical resource allocation. Third, ENS-MOEA/D and MOEA/D-FRRMAB also implement

[1] The code of NSGA-II is from https://www.egr.msu.edu/~kdeb/codes.shtml.
[2] The codes of SMPSO and SMS-EMOA are available at https://sourceforge.net/projects/jmetal/?source=directory.
[3] The code of HypE is from http://www.tik.ee.ethz.ch/pisa.
[4] The codes of MOEA/D-DE and MOEA/D-DRA are from http://dces.essex.ac.uk/staff/zhang/webofmoead.htm.
[5] The code of ENS-MOEA/D can be downloaded from http://www.ntu.edu.sg/home/epnsugan/.
[6] The code of MOEA/D-FRRMAB is available at https://github.com/Jerryl00/releasing-codes-java.

Table 3
IGD-metric comparisons of the algorithms for DTLZ problems.

	MS-DABC	NSGA-II	SMPSO	SMS-EMOA	HypE	MOEA/D-DE	MOEA/D-DRA	ENS-MOEA/D	FRRMAB
DTLZ1	2.45e-2	3.12e-2	3.00e-2	2.51e-2	1.69e-1	2.57e-2	2.52e-2	**2.13e-2**	2.13e-2
	(8.84e-4)3	(7.60e-3)8+	(1.39e-3)7+	(2.05e-4)4=	(3.84e-2)9+	(1.40e-3)6=	(2.75e-4)5=	**(1.26e-5)1-**	(8.21e-4)2-
DTLZ2	**5.14e-2**	6.98e-2	7.28e-2	6.97e-2	5.57e-2	6.94e-2	6.96e-2	6.44e-2	5.28e-2
	(1.17e-5)1	(2.53e-3)8+	(3.93e-3)9+	(5.52e-4)7+	(1.72e-3)3+	(6.38e-4)5+	(8.05e-4)6+	(2.67e-3)4+	(1.24e-3)2+
DTLZ3	**5.39e-2**	2.41e-1	7.73e-2	2.36e-1	5.97e-1	4.47e-1	7.18e-2	9.41e-2	1.67e-1
	(2.34e-3)1	(3.36e-1)7+	(1.72e-2)3+	(4.62e-1)6+	(4.82e-3)9+	(7.27e-1)8+	(3.73e-3)2+	(6.89e-2)4+	(1.69e-1)5+
DTLZ4	**5.14e-2**	6.76e-2	7.09e-2	1.99e-1	3.55e-1	6.99e-2	6.79e-2	6.35e-2	5.26e-2
	(1.11e-5)1	(2.00e-3)4+	(3.80e-3)7+	(1.40e-1)8+	(3.46e-1)9+	(3.08e-3)6+	(8.16e-4)5+	(2.35e-3)3+	(2.18e-3)2=
DTLZ5	4.13e-3	5.46e-3	**3.95e-3**	1.25e-2	1.15e-2	1.25e-2	1.25e-2	3.29e-2	4.36e-3
	(4.27e-5)2	(2.32e-4)4+	**(5.86e-5)1-**	(6.18e-5)8+	(7.25e-4)5+	(8.53e-5)7+	(1.10e-4)6+	(1.52e-3)9+	(1.31e-4)3+
DTLZ6	4.07e-3	1.97e-2	2.40e-1	1.19e-2	2.19e-2	1.19e-2	1.19e-2	**3.95e-3**	4.23e-3
	(8.21e-5)2	(1.40e-2)7+	(2.65e-2)9+	(1.31e-5)5+	(2.07e-3)8+	(1.01e-5)6+	(5.99e-5)4+	**(1.02e-4)1-**	(2.54e-4)3=
DTLZ7	**7.32e-2**	7.65e-2	8.66e-2	2.08e-1	1.08e-1	1.84e-1	1.93e-1	1.21e-1	1.65e-1
	(4.87e-3)1	(4.02e-3)2=	(7.12e-3)3+	(7.40e-2)9+	(1.02e-1)4=	(9.82e-3)7+	(9.39e-3)8+	(2.48e-3)5+	(1.46e-1)6=
Rank sum	11	40	39	47	47	45	36	27	23
Mean Rank	1.5714	5.7143	5.5714	6.7143	6.7143	6.4286	5.1429	3.8571	3.2857
Final Rank	1	6	5	8	9	7	4	3	2
w/s/l		6/1/0	6/0/1	6/1/0	6/1/0	6/1/0	6/1/0	5/0/2	3/3/1

the co-evolving paradigms (e.g., the ensemble of multiple parameters and adaptive operator selection) to coordinate the same search as ours. Therefore, it is meaningful to conduct a comparison among them.

5.2.1. Comparisons on the DTLZ instances

The experimental results of the compared MOEAs for DTLZ problems are listed in Table 3, where the mean and standard deviation (in parentheses) of the IGD metric values for 30 runs are recorded. The numbers next to the parentheses refer to the rankings of algorithms for that problem, which are obtained based on the average metric value, and the best result is highlighted in bold type with a dark gray background, and the second best is highlighted in light gray background. The symbols "+", "=" and "−", respectively indicate that MS-DABC performs statistically better than, similar to and worse than the compared algorithms based on Wilcoxon's rank sum test. The last row denotes that MS-DABC wins on w problems, ties on t problems, and loses on l problems, compared with its corresponding competitor. For simplicity, the MOEA/D- FRRMAB is referred as FRRMAB for short. From Table 3, it can be seen that MS-DABC has the best IGD metric value in four cases while ENS-MOEA/D performs the best in two cases. However, the final rank implies that MOEA/D-FRRMAB achieves the second best while ENS-MOEA/D is placed third. This is due to MOEA/D-FRRMAB producing more stable and better results on DTLZ2 and DTLZ4 than ENS-MOEA/D. Another promising algorithm, MOEA/D-DRA, being ranked slightly lower than ENS-MOEA/D, achieves fourth place. Regarding the comparison results of Wilcoxon's rank sum test given in the last row, MS-DABC is superior to NSGA-II, SMPSO, SMS-EMOA, HypE, MOEA/D-DE, MOEA/D-DRA, ENS-MOEA/D, and MOEA/D-FRRMAB respectively with 6, 6, 6, 6, 6, 6, 5, and 3 wins out of 7 problems, only beaten by SMPSO on DTLZ5, ENS-MOEA/D on DTLZ1 and DTLZ6, and MOEA/D-FRRMAB on DTLZ1. It is also noted that variants of MOEA/D such as MOEA/D-DRA, ENS-MOEA/D and MOEA/D-FRRMAB perform better than MOEA/D-DE, especially on complicated problems such as DTLZ1 and DTLZ3 with multimodality, and DTLZ4 with bias landscapes. The reason may be due to the synergetic mechanism of multiple strategies in these variants, which is suitable for different types of search landscapes.

Fig. 5 presents the boxplots for SPREAD metric values of solutions obtained by the compared MOEAs over 30 independent runs. Table 4 gives the Statistics of SPREAD comparisons of MS-DABC with the other rivals, according to the Wilcoxon's rank sum test. Clearly, these results indicate that MS-DABC performs significantly better than its competitors on 47 out of 56 comparisons in terms of SPREAD metric. While MS-DABC is only outperformed by its rivals on 2 out of 56 comparisons. To be specific, MS-DABC is defeated by ENS-MOEA/D and MOEA/D- FRRMAB on DTLZ1 and DTLZ3, respectively. The superiority of MS-DABC is more obvious on complex problems with irregular PFs, e.g., DTLZ5-7.

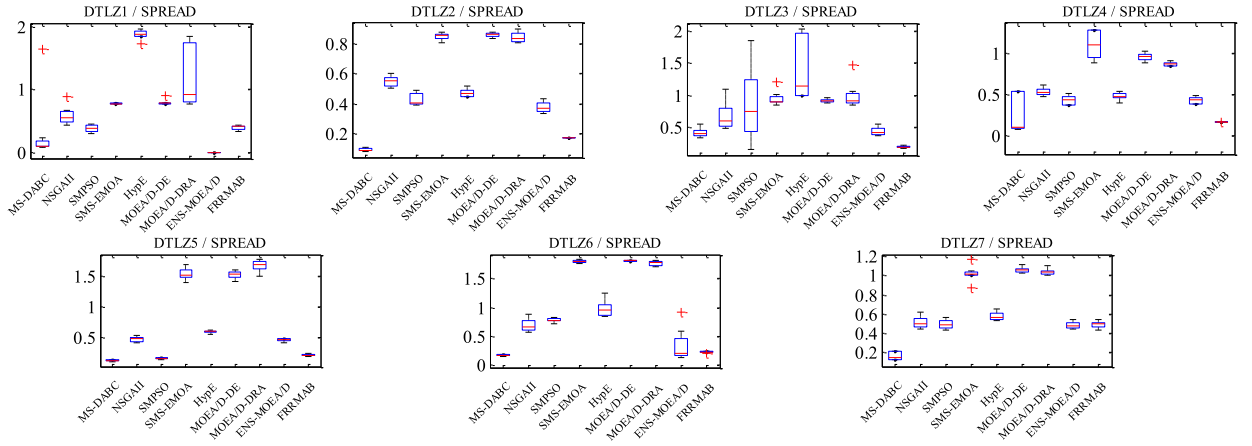

Fig. 5. Boxplots for SPREAD comparison of MOEAs on DTLZ problems.

Table 4
Statistics of SPREAD comparisons on DTLZ problems.

	NSGA-II	SMPSO	SMS-EMOA	HypE	MOEA/D-DE	MOEA/D-DRA	ENS-MOEA/D	FRRMAB
+	7	5	7	6	7	7	3	5
=	0	2	0	1	0	0	3	1
−	0	0	0	0	0	0	1	1

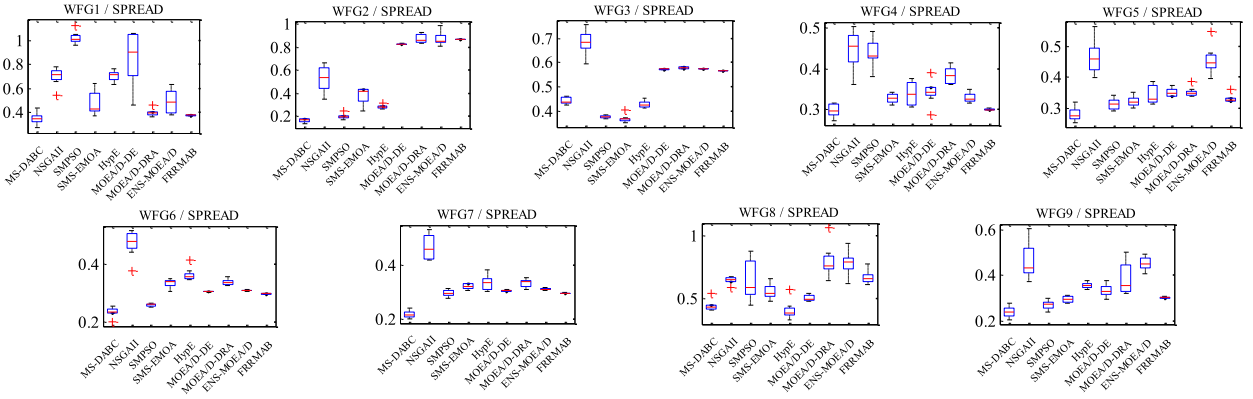

Fig. 6. Boxplots for SPREAD comparison of MOEAs on WFG problems.

5.2.2. Comparisons on the WFG instances

Comparison results of WFG benchmarks on IGD metric are presented in Table 5, from which we can observe that MS-DABC yields 7 out of 9 best metric values, while SMPSO and HypE win only 1 test case, respectively. Regarding the Wilcoxon's rank sum test, compared with the other eight MOEAs, MS-DABC achieves significant better result in 61 out of 72 IGD comparisons. In terms of overall final rank, MS-DABC is ranked the best, followed by SMS-EMOA and MOEA/D-FRRMAB. These observations demonstrate the advantages of MS-DABC in tackling WFG problems. It is interesting to note that two indicator based algorithms, i.e., SMS-EMOA and HypE, perform well on WFG instances, ranked second and fourth overall in this comparison. It is also observed that, by using some adaptive strategies (e.g., dynamical resource allocation for MOEA/D-DRA, adaptive neighborhood size for ENS-MOEA/D), the modified algorithms perform better than their standard version MOEA/D-DE. MS-DABC and MOEA/D-FRRMAB share some similarities in maintaining diversity with the help of adaptive operator selection, and they both show promising performance in addressing WFG instances.

Fig. 6 illustrates the boxplots for SPREAD metric values over 30 runs obtained by MS-DABC and its competitors, from which it can be observed that MS-DABC performs either comparably or remarkably better than its rivals on most test instances except WFG3, where it is beaten by SMPSO and SMS-EMOA. The SPREAD metric comparisons of MS-DABC with its rivals, regarding Wilcoxon's rank sum test, are presented in Table 6, which reveals that MS-DABC achieves significant better results in 67 out of 72 comparisons. Especially, MS-DABC significantly outperforms MOEA/D-FRRMAB in 8 out of 9 cases.

Table 5
IGD-metric comparisons of the algorithm for WFG problems.

	MS-DABC	NSGA-II	SMPSO	SMS-EMOA	HypE	MOEA/D-DE	MOEA/D-DRA	ENS-MOEA/D	FRRMAB
WFG1	**1.44e-2**	4.04e-2	1.06e+0	3.72e-1	7.65e-1	2.41e-1	5.51e-2	5.13e-2	1.11e-1
	(4.42e-4)1	(2.24e-2)2+	(1.52e-1)9+	(3.53e-1)7+	(1.33e-1)8+	(1.19e-1)6+	(8.36e-2)4+	(6.67e-2)3+	(1.16e-1)5+
WFG2	1.33e-2	1.27e-2	**1.02e-2**	1.13e-2	1.07e-2	3.68e-2	3.67e-2	3.64e-2	1.05e-2
	(5.05e-4)6	(6.81e-4)5=	**(8.51e-4)1-**	(4.83e-4)4-	(4.89e-4)3-	(4.37e-4)9+	(9.01e-5)8+	(2.02e-3)7+	(5.13e-4)2-
WFG3	**4.36e-3**	1.53e-2	1.17e-2	1.14e-2	1.18e-2	1.32e-2	1.32e-2	1.32e-2	1.17e-2
	(4.76e-7)1	(6.38e-4)9+	(1.10e-4)4+	(2.76e-4)2+	(2.00e-4)5+	(2.44e-5)8+	(1.07e-5)7+	(5.09e-6)6+	(8.99e-5)3+
WFG4	**5.06e-3**	1.37e-2	2.57e-2	1.00e-2	1.02e-2	1.90e-2	1.60e-2	1.60e-2	1.09e-2
	(9.38e-6)1	(4.51e-4)5+	(1.81e-3)9+	(2.06e-4)2+	(3.35e-4)3+	(3.27e-3)8+	(3.69e-4)7+	(3.63e-4)6+	(2.61e-4)4+
WFG5	**6.32e-2**	6.66e-2	6.64e-2	6.65e-2	6.67e-2	6.72e-2	6.70e-2	6.91e-2	6.66e-2
	(5.72e-3)1	(4.60e-3)5+	(1.04e-4)2+	(5.47e-5)3+	(1.17e-4)6+	(3.89e-4)8+	(1.58e-4)7+	(7.44e-4)9+	(1.41e-4)4+
WFG6	**4.96e-3**	2.65e-2	1.28e-2	2.23e-2	6.36e-2	1.54e-2	1.54e-2	1.54e-2	1.41e-2
	(4.65e-6)1	(1.27e-2)8+	(1.70e-4)2+	(9.11e-3)7+	(1.18e-2)9+	(2.37e-5)5+	(6.98e-5)6+	(2.30e-5)4+	(2.21e-3)3+
WFG7	**5.43e-3**	1.58e-2	1.19e-2	1.08e-2	1.10e-2	1.63e-2	1.62e-2	1.62e-2	1.24e-2
	(1.23e-6)1	(6.71e-4)6+	(1.18e-4)4+	(3.22e-4)2+	(2.88e-4)3+	(2.65e-5)9+	(2.54e-5)8+	(7.58e-6)7+	(2.10e-4)5+
WFG8	1.08e-1	1.46e-1	1.54e-1	1.55e-1	**9.87e-2**	1.49e-1	1.37e-1	1.34e-1	1.85e-1
	(7.97e-2)2	(4.77e-2)5=	(3.14e-2)7=	(4.51e-2)8+	**(4.04e-3)1=**	(4.90e-2)6=	(6.72e-2)4=	(5.64e-2)3=	(1.04e-2)9+
WFG9	**4.98e-3**	1.72e-2	1.55e-2	1.19e-2	1.42e-2	1.65e-2	1.52e-2	1.61e-2	1.31e-2
	(2.28e-4)1	(1.99e-3)9+	(6.24e-4)6+	(6.75e-4)2+	(7.19e-4)4+	(1.05e-3)8+	(2.00e-4)5+	(2.41e-4)7+	(8.50e-4)3+
Rank sum	15	54	44	37	42	67	56	52	38
Mean Rank	1.6667	6	4.8889	4.1111	4.6667	7.4444	6.2222	5.7778	4.2222
Final Rank	1	7	5	2	4	9	8	6	3
w/s/l		7/2/0	7/1/1	8/0/1	7/1/1	8/1/0	8/1/0	8/1/0	8/0/1

Table 6
Statistics of SPREAD comparisons on WFG problems.

	NSGA-II	SMPSO	SMS-EMOA	HypE	MOEA/D-DE	MOEA/D-DRA	ENS-MOEA/D	FRRMAB
+	9	8	8	7	9	9	9	8
=	0	0	0	2	0	0	0	1
−	0	1	1	0	0	0	0	0

This implies that, besides adaptive operator selection, the well maintained *EXA* and potential region detection of onlookers could be other important factors for the better distribution of solutions obtained by MS-DABC.

5.3. The scalability of MS-DABC for tackling MaOPs

In order to investigate the scalability of our proposed algorithm in tackling problems with many objectives, MS-DABC is compared with some state-of-the-art algorithms on DTLZ1 to DTLZ4 of the DTLZ benchmark problems, which can be scaled to an arbitrary number of objectives and decision variables. To be more specific, the number of objectives for each DTLZ instance varies from 5 to 15, i.e., $m \in \{5, 8, 10, 13, 15\}$, and the number of decision variables is set as $n = m + r - 1$, where r is set to 5 for DTLZ1 and 10 for the rest test problems. To calculate IGD, the approach in [20] is employed to generate the

Table 7
Setting of the population size.

Objective number (m)	Divisions (H)	Population size (N)
5	6, 0	210
8	3, 2	156
10	3, 2	275
13	2, 2	182
15	2, 1	135

reference PFs. Four popular MOEAs for many-objective optimization, KnEA[7] [46], NSGA-III[8] [5], SPEA/R[7] [12], and MOEA/DD[9] [20], are considered for comparison. KnEA is a knee point-driven evolutionary algorithm for solving MaOPs. NSGA-III is an improved version of NSGA-II that uses niche-preservation operation to rescue the loss of selection pressure caused by dominance resistant solutions in a high-dimensional objective space. SPEA/R is a recently proposed reference vector guided algorithm with encouraging performance. Since MOEA/DD also uses a hybrid framework of decomposition and domination similar to MS-DABC, it is interesting to make a comparison of them. The parameters of the considered algorithms are set as suggested in the original literature.

Due to the combinatorial nature for the generation mode of the weight vectors, the population size cannot be arbitrarily set. The settings of population size N for problems with a varying number of objectives are listed in Table 7. Noting that the double-layer weight vector generation method is employed in the case of $m > 5$. The termination criterion of each algorithm is the prespecified *Max_FEs*, which is set as 600**N*, 750**N*, 1000**N*, 1200**N*, and 1500**N* for 5-, 8-, 10-, 13-, and 15-objective cases respectively based on their actual convergence conditions. In addition, each algorithm is repeated in 30 independent runs on each test instance.

The statistical results of IGD values for the compared algorithms on 20 MaOPs are presented in Table 8, where the best results in terms of mean metric values are highlighted in boldface. Meanwhile, the last row summarizes the problem number in which MS-DABC is significantly better than, similar with, and inferior to its competitors. From Table 8, we can see that MS-DABC shows best overall performance on the test instances according to the final rank summarized in the second last row. It seems that MOEA/DD also has a competitive overall performance, obtaining a slightly lower final rank than MS-DABC, and KnEA performs worst when all the test problems are considered. However, KnEA outperforms MS-DABC and competes well with MOEA/DD for the DTLZ2 with 5 and 8-objective cases. Additionally, Wilcoxon's rank sum test reveals that MS-DABC is superior to KnEA, NSGA-III, MOEA/DD, and SPEA/R on 18, 17, 12 and 20 out of 20 problems, respectively.

It is also observed that the performance of the algorithms has a slight difference with varying problem characteristics and the objective number. For DTLZ1, a hard-to-converge problem, we find that MOEA/DD outperforms the others for the case of 13 objectives, while MS-DABC has the best performance in other cases. For DTLZ2, a relatively simple problem, all the competitors can achieve promising performance in the 5 and 8-objective cases. To be specific, KnEA wins in the 5-objective case, and MOEA/DD can compete with MS-DABC in the 8 and 10-objective cases. However, MS-DABC has the best IGD values when the number of objectives is over 10. For DTLZ3, a highly multimodal problem with many local optima, MOEA/DD is the best algorithm in the 5, 8, and 10-objective instances. However, the situation changes for the cases with more than 10 objectives, where MOEA/DD is defeated by MS-DABC. For DTLZ4, characterized by a strongly biased PF, MS-DABC performs the best in all cases. Overall, MS-DABC performs better than MOEA/DD despite their similarity in maintaining diversity with the aid of weight vectors. This may be attributed to the new fitness assignment and synergy of the operators in the former.

Table 9 presents the SPREAD metric values obtained by the compared algorithms. Clearly, MS-DABC gives the best results, in terms of mean SPREAD values, on 9 out of 20 test instances, performing the best among all the competing algorithms. Specifically, its good performance is noticeable on problems with multi-model features, such as DTLZ1 and DTLZ3. Regarding the statistical comparison a result of Wilcoxon's rank sum test, MS-DABC is, respectively, better than KnEA, NSGA-III, MOEA/DD and SPEA/R on 10, 9, 10, 11 out of 20 test instances. In total, it performs significantly better and worse than its competitors in 40 and 13 out of 80 comparisons, respectively. As for the overall performance ranking, MS-DABC is placed at the top among all competitors. These observations confirm the advantage of MS-DABC in achieving well- distributed solutions along the desired PF.

To take a closer look at the distribution of PFs obtained by the algorithms, we plot the parallel coordinates of the final PFs with the median IGD values obtained by each algorithm on DTLZ1 to DTLZ4 instances in Figs. 7–10, respectively, among which Figs. 7 and 8 are associated with the 15-objective case, while Figs. 9 and 10 are associated with the 13-objective case.

As can be seen from Fig. 7, the PFs obtained by MS-DABC, NSGA-III and MOEA/DD show promising convergence and diversity on the 15-objective DTLZ1, whereas both KnEA and SPEA/R fail to converge to the true PF distributed over $f_i \in [0, 0.5]$. Comparing these two algorithms, KnEA has a more uniform distribution of solutions, while SPEA/R only concentrates on certain parts of the PF. For the 15-objective DTLZ2, presented in Fig. 8, all the algorithms can achieve encouraging performance except NSGA-III, which fails to approximate well on the 11th to 15th objectives. The results of the 13-objective

[7] The codes of KnEA and SPEA/R were provided by their authors.
[8] The code of NSGA-III is from http://web.ntnu.edu.tw/~tcchiang/publications/nsga3cpp/nsga3cpp.htm.
[9] The code of MOEA/DD is available at https://github.com/Jerryl00/releasing-codes-java.

Table 8
IGD-metric comparisons for many-objective DTLZ problems.

	m	MS-DABC	KnEA	NSGA-III	MOEA/DD	SPEA/R
DTLZ1	5	**2.66** e − **4(6.40** e − **5)1**	2.07 e − 1(9.06 e − 2)5+	5.27 e − 4(2.31 e − 4)2+	5.96 e − 4(1.92 e − 4)3+	6.07 e − 2(5.20 e − 2)4+
	8	**3.05** e − **3(6.66** e − **4)1**	2.98 e − 1(1.10 e − 1)5+	7.15 e − 3(6.20 e − 3)3+	4.13 e − 3(8.72 e − 4)2+	1.30 e − 1(3.27 e − 2)4+
	10	**2.77** e − **3(6.57** e − **4)1**	4.20e + 0(4.41 e + 0)5+	3.38 e − e 2(8.19 e − 2)3+	3.12 e − 3(3.37 e − 4)2=	1.83 e − 1(2.82 e − 2)4+
	13	4.95 e − e 2(2.82 e − 3)3	6.92e + 0(4.14 e + 0)5+	3.84 e − 2(7.58 e − 2)2-	**1.66** e − **2(9.99** e − **3)1-**	2.96 e − 1(6.89 e − 2)4+
	15	**4.30** e − **3(1.47** e − **3)1**	5.85e + 0(6.98 e + 0)5+	2.30 e − 2(5.24 e − 2)3=	1.04 e − 2(7.93 e − 3)2+	3.01 e − 1(8.07 e − 2)4+
DTLZ2	5	1.63 e − 3(2.20 e − 4)3	**1.21** e − **3(8.27** e − **5)1-**	2.29 e − 3(3.39 e − 4)4+	1.25 e − 3(7.92 e − 5)2-	3.94 e − 3(1.35 e − 3)5+
	8	5.39 e − 3(4.08 e − 4)3	3.23 e − 3(2.56 e − 4)2-	9.45 e − 3(5.89 e − 4)4+	**3.17** e − **3(3.64** e − **4)1-**	1.90 e − 2(2.90 e − 3)5+
	10	5.70 e − 3(7.91 e − 4)2	3.25 e − 2(9.71 e − 4)4+	5.98 e − 2(1.46 e − 1)5+	**4.13** e − **3(2.63** e − **4)1-**	2.44 e − 2(2.98 e − 3)3+
	13	**1.16** e − **2(8.12** e − **4)1**	7.41 e − 2(3.68 e − 3)3+	2.08 e − 1(2.00 e − 1)5+	8.29 e − 2(5.44 e − 3)4+	3.38 e − 2(8.17 e − 3)2+
	15	**4.63** e − **3(2.63** e − **4)1**	1.73 e − 2(2.31 e − 3)3+	2.73 e − 1(1.24 e − 1)5+	7.05 e − 3(4.97 e − 4)2+	4.44 e − 2(7.14 e − 3)4+
DTLZ3	5	2.18 e − 3(1.44 e − 3)2	5.82 e − 1(1.54 e − 1)5+	2.58 e − 3(9.20 e − 4)3=	**1.90** e − **3(1.11** e − **3)1=**	5.37 e − 1(2.66 e − 1)4+
	8	8.51 e − 3(1.28 e − 3)2	4.21e + 1(1.68 e + 1)5+	1.72 e − 1(2.19 e − 1)3+	**7.63** e − **3(3.51** e − **3)1=**	4.22e + 0(3.31 e + 0)4+
	10	5.74 e − 3(7.70 e − 4)2	3.20e + 2(8.26 e + 1)5+	7.64 e − 2(1.84 e − 1)3+	**4.53** e − **3(1.11** e − **3)1-**	8.86e + 0(2.26 e + 0)4+
	13	**1.90** e − **2(1.31** e − **3)1**	4.10e + 2(1.21 e + 2)5+	7.96 e − 1(8.63 e − 1)3+	9.49 e − 2(1.05 e − 2)2+	7.79e + 0(5.88 e + 0)4+
	15	**4.97** e − **3(1.15** e − **3)1**	5.43e + 2(1.58 e + 2)5+	7.87 e − 1(6.93 e − 1)3+	1.13 e − 2(2.53 e − 3)2+	1.29e + 0(1.15 e + 0)4+
DTLZ4	5	**1.14** e − **4(6.17** e − **6)1**	1.18 e − 1(8.75 e − 3)5+	1.26 e − 3(5.21 e − 4)3+	3.00 e − 4(3.50 e − 5)2+	6.42 e − 3(9.38 e − 4)4+
	8	**6.38** e − **4(9.90** e − **5)1**	2.53 e − 1(1.65 e − 2)5+	9.17 e − 2(1.63 e − 1)4+	3.04 e − 3(3.70 e − 4)2+	1.61 e − 2(1.99 e − 3)3+
	10	**1.48** e − **3(2.51** e − **4)1**	2.55 e − 1(7.58 e − 3)5+	3.58 e − 3(3.88 e − 4)3+	3.34 e − 3(1.39 e − 4)2+	2.31 e − 2(2.32 e − 3)4+
	13	**1.21** e − **3(1.08** e − **4)1**	3.09 e − 1(1.93 e − 2)5+	1.71 e − 1(2.27 e − 1)4+	5.23 e − 3(6.91 e − 4)2+	1.75 e − 2(2.50 e − 3)3+
	15	**2.18** e − **3(6.22** e − **4)1**	2.37 e − 1(7.84 e − 3)5+	1.68 e − 1(2.26 e − 1)4+	3.09 e − 3(2.67 e − 4)2+	2.97 e − 2(4.67 e − 3)3+
Rank sum		30	88	69	37	76
Mean Rank		1.5	4.4	3.45	1.85	3.8
Final Rank		1	5	3	2	4
w/s/l			18/0/2	17/2/1	12/3/5	20/0/0

Table 9
SPREAD-metric comparisons for many-objective DTLZ problems.

	m	MS-DABC	KnEA	NSGA-III	MOEA/DD	SPEA/R
DTLZ1	5	**2.01 e − 3(9.80 e − 4)1**	1.11 e + 0(4.41 e − 1)4+	3.09 e − 3(5.75 e − 4)2+	4.87 e − 3(2.13 e − 3)3+	1.12 e + 0(8.40 e − 1)5+
	8	**6.74 e − 2(3.02 e − 3)1**	1.41 e + 0(2.49 e − 1)4+	9.17 e − 2(3.94 e − 2)3+	7.87 e − 2(5.81 e − 3)2+	1.81 e + 0(1.03 e − 1)5+
	10	**6.04 e − 2(2.76 e − 3)1**	3.61 e − 1(1.60 e − 1)4+	3.28 e − 1(6.39 e − 1)3=	6.37 e − 2(1.80 e − 3)2+	1.74 e + 0(8.61 e − 2)5+
	13	1.70 e + 0(4.49 e − 2)5	3.87 e − 1(1.88 e − 1)3-	3.10 e − 1(5.90 e − 1)2-	**1.05 e − 1(3.07 e − 2)1-**	1.62 e + 0(7.43 e − 2)4-
	15	**1.94 e − 2(1.07 e − 2)1**	2.92 e − 1(1.53 e − 1)3+	3.06 e − 1(6.63 e − 1)4+	9.64 e − 2(1.39 e − 2)2+	1.43 e + 0(9.24 e − 2)5+
DTLZ2	5	1.51 e − 1(2.01 e − 4)2	2.30 e − 1(8.58 e − 2)5+	**1.51 e − 1(2.88 e − 4)1=**	1.51 e − 1(1.91 e − 4)3=	1.51 e − 1(1.21 e − 3)4=
	8	1.36 e − 1(1.97 e − 3)3	2.35 e − 1(1.78 e − 1)5=	1.36 e − 1(4.66 e − 3)2=	1.39 e − 1(6.76 e − 4)4+	**1.33 e − 1(4.36 e − 3)1=**
	10	1.52 e − 1(1.50 e − 3)3	**1.10 e − 1(9.81 e − 2)1=**	2.11 e − 1(1.65 e − 1)5=	1.51 e − 1(1.46 e − 3)2=	1.52 e − 1(2.56 e − 3)4=
	13	**5.04 e − 2(2.36 e − 3)1**	5.35 e − 2(2.08 e − 2)2=	3.71 e − 1(3.33 e − 1)4+	4.77 e − 1(2.44 e − 2)5+	8.91 e − 2(9.14 e − 2)3+
	15	**4.79 e − 2(2.91 e − 3)1**	1.40 e − 1(1.72 e − 1)3=	5.32 e − 1(2.07 e − 1)5+	3.25 e − 1(2.01 e − 3)4+	5.54 e − 2(5.68 e − 3)2+
DTLZ3	5	1.51 e − 1(1.88 e − 4)3	1.58 e + 0(4.75 e − 1)5+	1.51 e − 1(3.91 e − 4)2=	**1.51 e − 1(1.13 e − 4)1-**	1.40 e + 0(1.35 e − 1)4+
	8	**1.35 e − 1(1.00 e − 3)1**	5.13 e − 1(1.23 e − 1)3+	7.28 e − 1(8.14 e − 1)4+	1.40 e − 1(8.19 e − 4)2+	1.10 e + 0(1.20 e − 1)5+
	10	1.52 e − 1(5.20 e − 4)2	3.50 e − 1(1.19 e − 1)3+	3.73 e − 1(6.16 e − 1)4+	**1.50 e − 1(1.31 e − 3)1=**	9.26 e − 1(7.53 e − 2)5+
	13	**5.59 e − 2(2.75 e − 3)1**	2.29 e − 1(7.78 e − 2)2+	1.66 e + 0(1.45 e − 1)5+	1.64 e + 0(8.30 e − 2)4+	1.01 e + 0(7.14 e − 2)3+
	15	**4.90 e − 2(1.25 e − 3)1**	1.68 e − 1(4.23 e − 2)2+	1.66 e + 0(1.16 e − 1)5+	3.25 e − 1(2.37 e − 3)3+	1.03 e + 0(5.07 e − 2)4+
DTLZ4	5	1.51 e − 1(2.40 e − 5)3	2.31 e − 1(1.56 e − 1)5=	**1.51 e − 1(3.85 e − 4)1=**	1.51 e − 1(2.45 e − 5)2−	1.51 e − 1(1.79 e − 3)4=
	8	1.40 e − 1(3.23 e − 4)3	1.40 e − 1(1.26 e − 1)4=	2.68 e − 1(2.42 e − 1)5=	1.37 e − 1(1.11 e − 3)2−	**1.35 e − 1(2.12 e − 3)1−**
	10	1.52 e − 1(1.11 e − 3)3	**6.83 e − 2(3.03 e − 2)1-**	1.53 e − 1(7.74 e − 4)5=	1.53 e − 1(1.02 e − 3)4=	1.52 e − 1(2.42 e − 3)2=
	13	5.20 e − 2(1.45 e − 3)3	6.28 e − e 2(1.73 e − 2)4=	2.78 e − 1(3.25 e − 1)5=	5.19 e − 2(2.18 e − 3)2=	**4.75 e − 2(2.69 e − 3)1-**
	15	3.18 e − 1(1.73 e − 3)5	7.56 e − 2(3.64 e − 2)3−	2.97 e − 1(3.61 e − 1)4=	**4.57 e − 2(1.56 e − 3)1-**	4.81 e − 2(4.89 e − 3)2−
Rank sum		44	66	71	50	69
Mean Rank		2.2	3.3	3.55	2.5	3.45
Final Rank		1	3	5	2	4
w/s/l			10/7/3	9/10/1	10/5/5	11/5/4

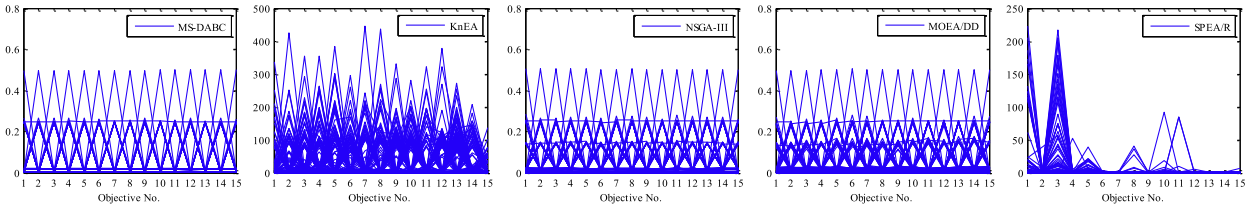

Fig. 7. Parallel coordinates of PF for 15-objective DTLZ1.

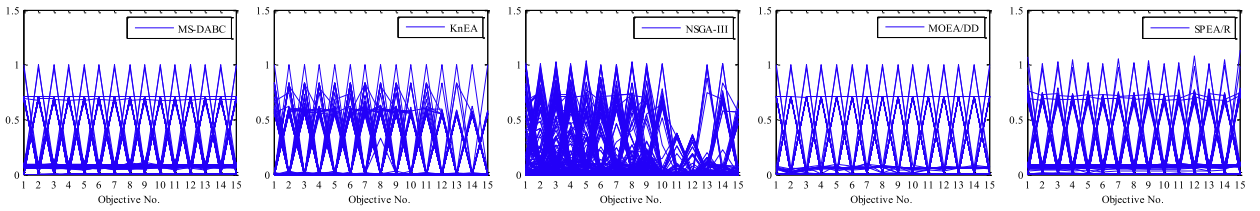

Fig. 8. Parallel coordinates of PF for 15-objective DTLZ2.

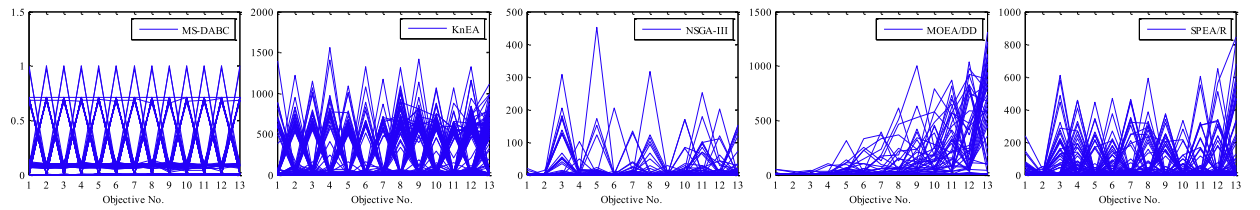

Fig. 9. Parallel coordinates of PF for 13-objective DTLZ3.

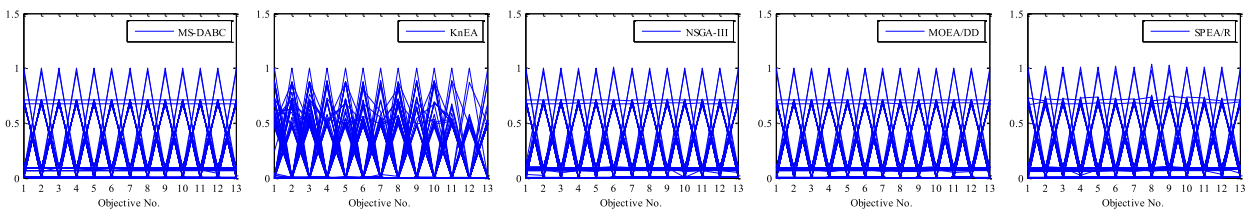

Fig. 10. Parallel coordinates of PF for 13-objective DTLZ4.

DTLZ3 are shown in Fig. 9. It is clear that MS-DABC is the only algorithm achieving a good distribution of solutions converging well for the entire PF, and other competitors fail to converge to the true PFs because their objective values of solutions are far from that of the true PF. Fig. 10 shows similar PF results to a certain extent in visualization, except that KnEA obtains slightly messy PFs. Despite that, the IGD results from Table 8 indicate that MS-DABC outperforms other competitors by a clear margin.

In addition, the chosen algorithms are tested on the WFG problems with five- and ten-objectives. To calculate the IGD metric, the reference PF is required. Since the PF of WFG4-9 is a hypersphere, we can easily get the reference PF by multiplying the ith objective of PF for DTLZ2-4. For WFG1-3 whose PF shapes are irregular, the method in [4] is suggested for constructing the reference PFs. The population sizes are respectively set to 126 and 220 in the 5- and 10-objective WFG problems. The termination criterion is set to 1000 iterations for the 5-objective problems and 1500 iterations for the 10-objective problems.

The statistical results in terms of the IGD metric are presented in Table 10. It can be observed that MS-DABC, achieving the 11 best results out of 18 test instances, ranks first place, followed by NSGA-III, which performs the best on five test instances, while MOEA/DD is not the best on any instance. The failure of MOEA/DD is mainly because it lacks a normalization procedure for tackling MaOPs with badly scaled PFs. According to the statistical results summarized in the last row of the table, MS-DABC performs significantly better than KnEA, NSGA-III, MOEA/DD and SPEA/R on 14, 10, 17 and 11 out of 18 test instances, respectively. Generally, MS-DABC obtained significantly better performance in 52 out of 72 comparisons with its competitors.

More specifically, MS-DABC exhibits excellent performance on WFG1 and WFG4, the former having bias and mixed PF shapes while the latter features a multimodal landscape. This indicates that MS-DABC is suitable for such types of problems. Besides, MS-DABC competes well on WFG2 and WFG3, taking the first or second place, which implies MS-DABC can handle

Table 10
IGD-metric comparisons for many-objective WFG problems.

	m	MS-DABC	KnEA	NSGA-III	MOEA/DD	SPEA/R
WFG1	5	**4.98 e − 1(3.26 e − 2)1**	6.36 e − 1(9.25 e − 2)2+	8.08 e − 1(3.65 e − 2)3+	3.61e + 0(1.20 e − 1)5+	8.41 e − 1(1.59 e − 2)4+
	10	**1.43 e + 0(2.50 e − 1)1**	1.94e + 0(8.03 e − 1)2=	2.41e + 0(2.25 e − 1)3+	9.94e + 0(5.01 e − 2)5+	2.56e + 0(1.56 e − 1)4+
WFG2	5	**6.09 e − 1(7.56 e − 2)1**	6.47 e − 1(1.68 e − 1)3=	6.40 e − 1(4.08 e − 2)2=	4.09e + 0(3.74 e − 1)5+	6.71 e − 1(1.36 e − 1)4=
	10	1.47 e + 0(2.97 e − 1)2	**1.21e + 0(2.33 e − 1)1=**	3.37e + 0(9.36 e − 1)4+	1.43e + 1(6.21 e − 2)5+	2.49e + 0(3.42 e − 1)3+
WFG3	5	**4.62 e − 1(3.97 e − 2)1**	5.84 e − 1(9.60 e − 2)4+	4.87 e − 1(6.49 e − 2)2=	6.26 e − 1(3.00 e − 2)5+	4.90 e − 1(3.64 e − 2)3=
	10	1.40 e + 0(1.02 e − 1)2	1.95e + 0(5.10 e − e 1)3+	**6.58 e − 1(1.67 e − 1)1-**	2.51e + 0(1.85 e − 1)5+	1.98e + 0(8.25 e − 2)4+
WFG4	5	**1.09 e + 0(1.24 e − 2)1**	1.17e + 0(1.17 e − e 2)4+	1.10e + 0(1.39 e − 3)2+	1.20e + 0(9.33 e − 4)5+	1.11e + 0(4.21 e − 4)3+
	10	**4.42 e + 0(3.49 e − 2)1**	4.77e + 0(3.05 e − 2)2+	4.78e + 0(8.31 e − 2)3+	6.66e + 0(3.19 e − 1)5+	4.84e + 0(1.60 e − 3)4+
WFG5	5	1.10 e + 0(9.26 e − 3)3	1.15e + 0(1.27 e − 2)4+	**1.09e + 0(1.69 e − 3)1-**	1.17e + 0(1.06 e − 3)5+	1.09e + 0(5.50 e − 4)2-
	10	**4.32 e + 0(4.90 e − 2)1**	4.72e + 0(5.20 e − 2)2+	4.78e + 0(1.35 e − 2)3+	6.78e + 0(9.20 e − 2)5+	4.80e + 0(4.18 e − 3)4+
WFG6	5	1.12 e + 0(1.07 e − 2)3	1.20e + 0(2.53 e − 2)5+	**1.09e + 0(2.31 e − 3)1−**	1.18e + 0(3.11 e − 3)4+	1.09e + 0(2.51 e − e − 3)2−
	10	**4.53 e + 0(5.60 e − e 2)1**	4.84e + 0(8.40 e − 2)4+	4.82e + 0(3.68 e − 3)2+	6.80e + 0(1.69 e − 1)5+	4.82e + 0(2.86 e − 3)3+
WFG7	5	1.11e + 0(1.05 e − 2)3	1.18e + 0(2.52 e − 2)4+	**1.11e + 0(1.10 e − 3)1=**	1.20e + 0(1.25 e − 3)5+	1.11e + 0(6.20 e − 4)2=
	10	**4.33e + 0(4.28 e − 2)1**	4.64e + 0(4.32 e − 2)2+	4.88e + 0(2.16 e − 1)4+	5.71e + 0(4.93 e − 1)5+	4.85e + 0(4.44 e − 4)3+
WFG8	5	1.27e + 0(3.97 e − 2)4	1.31e + 0(3.13 e − 2)5+	1.21e + 0(6.91 e − 2)3=	1.19e + 0(5.70 e − 3)2−	**1.14e + 0(2.21 e − 2)1−**
	10	**4.59e + 0(8.72 e − 2)1**	4.94e + 0(1.37 e − 1)3+	5.32e + 0(3.83 e − 1)4+	6.81e + 0(2.23 e − 1)5+	4.83e + 0(9.26 e − 3)2+
WFG9	5	1.10e + 0(1.90 e − 2)4	1.09e + 0(1.80 e − 2)3=	**1.06e + 0(1.84 e − 2)1−**	1.16e + 0(4.57 e − 3)5+	1.07e + 0(2.13 e − 2)2−
	10	**4.18e + 0(4.45 e − 2)1**	4.52e + 0(2.12 e − 2)3+	4.51e + 0(6.65 e − 2)2+	6.72e + 0(7.63 e − 2)5+	4.65e + 0(9.21 e − 3)4+
Rank sum		32	56	42	86	54
Mean Rank		1.7778	3.1111	2.3333	4.7778	3
Final Rank		1	4	2	5	3
w/s/l			14/4/0	10/4/4	17/0/1	11/3/4

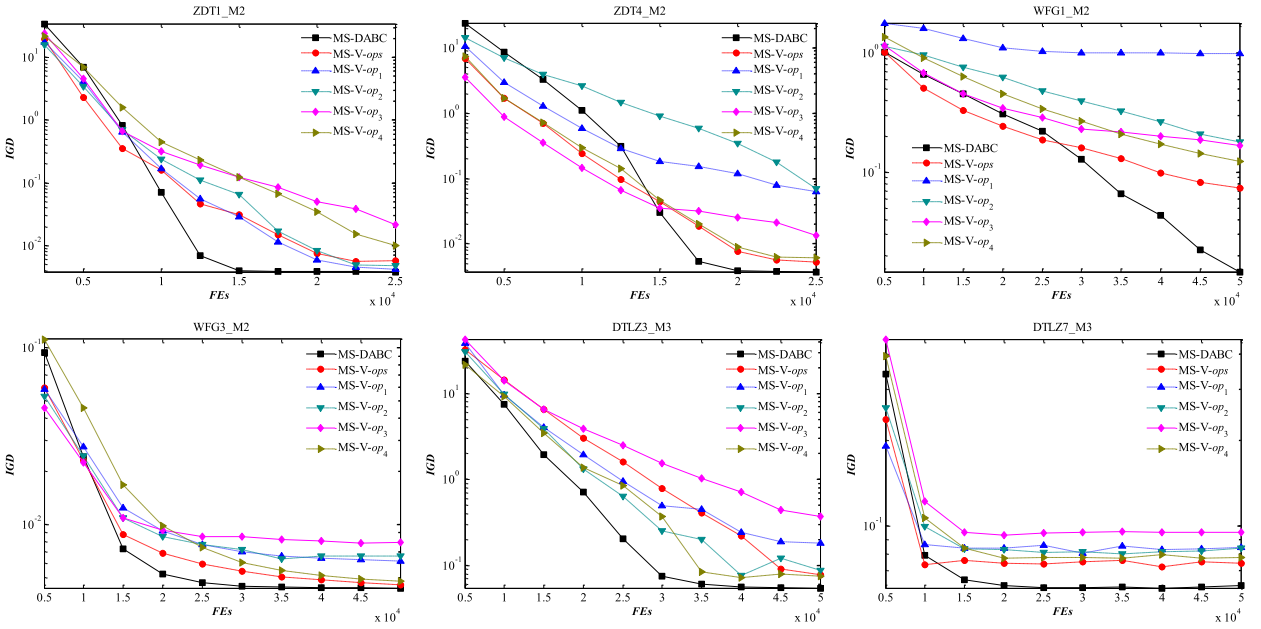

Fig. 11. Convergence performance of six algorithms over FEs on the test instances.

disconnected and degenerated PFs as well. As for WFG5-9 with identical PF surfaces but different problem characteristics, MS-DABC performs unsatisfactorily on problems with lower numbers of objectives, but achieves impressive performance for relatively high-dimensional cases. The superior performance of MS-DABC may be due to the prominent search direction detection of onlookers and dynamical resource allocation scheme.

Table 11 lists the comparison results of the five compared MaOEAs in terms of the SPREAD metric, among which MS-DABC has a clear advantage, followed by SPEA/R and NSGA-III, which are overall ranked at the second and third place, respectively, whereas MOEA/DD and KnEA struggle to maintain a good SPREAD metric. Thus, with consideration of the solution distribution, MS-DABC obtains the best performance. Overall, MS-DABC is very competitive on DTLZ and WFG test suite.

5.4. The effects of improvement strategies

The above experiments have clearly demonstrated the superiority of MS-DABC, but it is still not clear whether the modified strategies have a positive impact on the performance of the proposed algorithm. To answer such question, in this section, the effects of improvement strategies are exhaustively evaluated on the test instances. The experimental validation is divided into 3 case studies: (1) the first one is carried out to demonstrate the synergy effect of multiple operators; (2) the second one is used to investigate the effectiveness of the elite preserving *EXA* strategy; and (3) the last one is conducted to verify the impact of the fitness assignment mechanism.

5.4.1. Effectiveness of adaptive population size and multiple operator synergy

To investigate whether the incorporated strategies can help to improve the search performance of the proposed algorithm, we further design five MS-DABC variants for comparison. The first one, called MS-V-*ops*, is derived by removing the self-adaptive population adjustment mechanism from MS-DABC. The other four variants, i.e., MS-V-op_i ($i = 1,2,3,4$), are designed by employing only one evolutionary operator from Eqs. (19)–(22) in MS-DABC, respectively. Thus, MS-V-*ops* does not favor dynamical computational resource allocation among different operators whereas MS-V-op_i does not consider the synergy of multiple operators. The variants are compared with MS-DABC on 6 selected cases with the purpose of highlighting different observations. For ZDT problems, $N = 100$, and $Max\_FEs = 25{,}000$, all the algorithms are executed by 30 independent runs on each instance. To observe the search behavior of the compared algorithms, the mean IGD metric values for each test instance at 10 specified checkpoints are recorded.

The evolution of the mean IGD values against the numbers of function evaluations (FEs) for the compared MOEAs on 6 instances are presented in Fig. 11, where the speed and quality of convergence for each algorithm can be observed clearly. It can be seen that MS-DABC performs better than the others for all instances at the end of the search, despite struggling to reach a satisfactory convergence speed on most instances during the early evolutionary stage. For ZDT1 and WFG3, MS-DABC and MS-V-*ops* achieve a very close performance. Intuitively, MS-V-*ops* has the second best results, mostly. However, it is ranked the third worst in ZDT1 and defeated by MS-V-op_4 in DTLZ3, although showing good performance in the early

Table 11
SPREAD-metric comparisons for many-objective WFG problems.

	m	MS-DABC	KnEA	NSGA-III	MOEA/DD	SPEA/R
WFG1	5	**4.14 e − 1(5.54 e − 2)1**	6.89 e − 1(4.83 e − 2)5+	4.98 e − 1(1.80 e − 2)3+	6.01 e − 1(3.83 e − 2)4+	4.63 e − 1(1.63 e − 2)2+
	10	**5.98 e − 1(2.26 e − 2)1**	6.92 e − 1(4.20 e − 2)3+	7.11 e − 1(2.44 e − 2)4+	7.59 e − 1(1.47 e − 2)5+	6.79 e − 1(4.66 e − 2)2+
WFG2	5	**3.43 e − 1(3.83 e − 2)1**	5.23 e − 1(4.41 e − 2)5+	4.49 e − 1(6.82 e − 3)3+	4.34 e − 1(2.84 e − 2)2+	4.74 e − 1(3.80 e − 2)4+
	10	**5.10 e − 1(3.12 e − 2)1**	5.62 e − 1(3.96 e − 2)2+	7.53 e − 1(7.20 e − 2)5+	7.13 e − 1(4.39 e − 3)4+	6.39 e − 1(1.41 e − 2)3+
WFG3	5	**2.03 e − 1(8.75 e − 3)1**	2.98 e − 1(4.31 e − 2)2+	6.69 e − 1(6.39 e − 2)4+	3.62 e − 1(3.14 e − 2)3+	1.28e + 0(5.62 e − 2)5+
	10	**2.97 e − 1(1.69 e − 2)1**	3.80 e − 1(2.65 e − 2)3+	7.23 e − 1(1.76 e − 1)4+	3.06 e − 1(1.22 e − 2)2=	1.29e + 0(1.26 e − 1)5+
WFG4	5	**1.88 e − 1(2.06 e − 2)1**	3.69 e − 1(3.36 e − 2)5+	2.60 e − 1(2.85 e − 3)2+	3.55 e − 1(8.45 e − 4)4+	2.90 e − 1(3.29 e − 3)3+
	10	**2.60 e − 1(1.68 e − 2)1**	4.63 e − 1(3.26 e − 2)5+	3.93 e − 1(4.58 e − 2)3+	4.52 e − 1(2.35 e − 2)4+	3.65 e − 1(1.19 e − 3)2+
WFG5	5	**2.02 e − 1(1.29 e − 2)1**	3.74 e − 1(2.43 e − 2)5+	2.63 e − 1(4.34 e − 3)2+	3.58 e − 1(9.84 e − 4)4+	3.07 e − 1(5.12 e − 3)3+
	10	**2.68 e − 1(9.89 e − 3)1**	4.08 e − 1(2.56 e − 2)4+	3.68 e − 1(5.47 e − 3)2+	4.48 e − 1(7.26 e − 3)5+	3.70 e − 1(9.78 e − 4)3+
WFG6	5	**2.07 e − 1(2.10 e − 2)1**	4.96 e − 1(7.48 e − 2)5+	2.57 e − 1(1.49 e − 3)2+	3.57 e − 1(1.39 e − 3)4+	2.93 e − 1(5.42 e − 3)3+
	10	**2.90 e − 1(2.48 e − 2)1**	5.03 e − 1(6.06 e − 2)5+	3.67 e − 1(3.46 e − 4)2+	4.30 e − 1(2.19 e − 2)4+	3.67 e − 1(6.10 e − 4)3+
WFG7	5	**1.97 e − 1(1.68 e − 2)1**	3.93 e − 1(2.99 e − 2)5+	2.54 e − 1(1.18 e − 3)2+	3.55 e − 1(9.21 e − 4)4+	2.89 e − 1(4.38 e − 3)3+
	10	**2.49 e − 1(2.49 e − 2)1**	3.32 e − 1(2.09 e − 2)2+	5.65 e − 1(8.41 e − 2)5+	4.77 e − 1(1.63 e − 3)4+	3.65 e − 1(5.16 e − 4)3+
WFG8	5	**1.77 e − 1(1.84 e − 2)1**	6.33 e − 1(8.15 e − 2)5+	3.42 e − 1(7.93 e − 2)3+	3.62 e − 1(5.29 e − 3)4+	3.19 e − 1(1.72 e − 2)2+
	10	**2.85 e − 1(1.37 e − 2)1**	5.08 e − 1(5.88 e − 2)4+	5.08 e − 1(8.07 e − 2)5+	4.44 e − 1(3.22 e − 2)3+	3.66 e − 1(8.22 e − 4)2+
WFG9	5	**1.92 e − 1(2.14 e − 2)1**	3.60 e − 1(4.10 e − 2)4+	2.90 e − 1(1.77 e − 2)2+	4.04 e − 1(9.39 e − 3)5+	3.16 e − 1(1.21 e − 2)3+
	10	**2.26 e − 1(1.12 e − 2)1**	4.18 e − 1(2.84 e − 2)5+	4.00 e − 1(5.47 e − 2)3+	4.08 e − 1(1.62 e − 2)4+	3.74 e − 1(5.15 e − 3)2+
Rank sum		18	74	56	69	53
Mean Rank		1	4.1111	3.1111	3.8333	2.9444
Final Rank		1	5	3	4	2
w/s/l			18/0/0	18/0/0	17/1/0	18/0/0

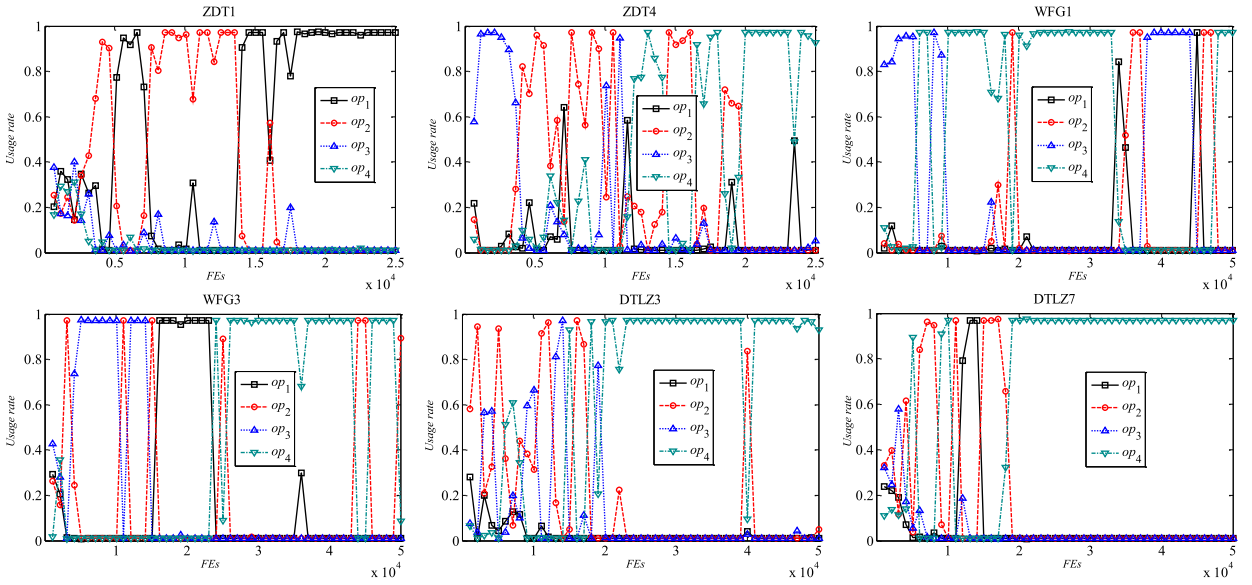

Fig. 12. Contributions of different search operators.

search process on ZDT1. Such results imply that the employment of multiple operators has positive effects on the algorithm but it cannot guarantee a promising convergence, especially in the final period of the searching process. This may be due to the fact that the diversity of the population is well retained during the entire search phase for MS-V-*ops*, which is good for exploring all the search space at the early stage, but is poor in further exploiting the identified promising regions at the latter stage. Therefore, only using multiple operators leads to overemphasis on the solution distribution and slows down the convergence rate. However, the *EXA* guided population size adjustment mechanism can determine an appropriate operator composition at different stages of evolution so as to fine tune the search, which guarantees a good convergence. Specifically for WFG1, a hard problem featured by mixed and biased PF, there is a big difference in the final IGD metric values, with MS-DABC showing significantly better convergence performance while the others seem to get stuck into local optima.

To provide more insights on the dynamics of an operator's performance over the evolutionary process, we identify and examine the application rates of different operators on selected problems (i.e., ZDT1, ZDT4, WFG1, WFG3, DTLZ3 and DTLZ7) corresponding to Fig. 11, and obtain the results shown in Fig. 12. Note that the application rate of op_i is represented by the portion of population size between $subpop_i$ and pop. From Fig. 12, we can observe that the application rate of an operator significantly varies during the evolutionary process. Moreover, the high performing operators for the proposed algorithm are problem dependent. For example in ZDT1, op_2 dominates the search at the early stage of evolution while it is outperformed by op_1 later. This observation is also confirmed in Fig. 11, where op_1 and op_2 are the top two best operators in solving ZDT1. Also, it is obvious that the exploration and exploitation search is carried out in a satisfactory manner: op_2 (i.e., DE/rand/2) with stronger exploration ability is performed at the early stage whereas op_1 (i.e., DE/rand/1) with a better exploitation ability is performed later. In addition, although op_2 and op_1 generally dominate the search process in ZDT1, they are rarely used and tend to serve as a foil to search for some unexplored regions of WFG1, where op_3 and op_4 play the main roles in guiding the evolutionary process. Similar situations can be observed in other cases.

One possible reason for this is that different problems have different landscapes in terms of search space, different operators have different levels of exploration and exploitation abilities, and some strategies are only suitable for specific problems. For a wide range of problems, using a single operator has the risk of loss of population diversity, which then implies a possibility of premature convergence. Employing multiple subpopulations which are assigned for different evolutionary operators working collaboratively may be a good choice. According to these results, it can be inferred that multiple operators can strengthen the search diversity while the *EXA* guided population size adjustment mechanism helps to guide the search direction for better convergence. Their joint efforts strengthen the robustness of MS-DABC.

5.4.2. The contribution of elite preserving strategy for EXA

In order to investigate whether the elite preserving strategy for *EXA* in MS-DABC affects the behavior of the proposed algorithm, we replace our strategy with other diversity management methods and derive two other MS-DABC variants for comparison. The first one, called MS-DABC-CD, adopts the crowding distance for *EXA* preservation. The second variant, MS-DABC-NP, employs niche preservation from NSGA-III for maintaining *EXA*.

Taking DTLZ6 and DTLZ7 with 3 objectives as examples, the final solutions obtained by the three algorithms are shown in Figs. 13 and 14, respectively. Clearly, MS-DABC with an elite preserving strategy for *EXA* has a competitive performance as it achieves a set of well converged and widely distributed solutions over the PF. For DTLZ6 with a degenerated PF, MS-DABC-

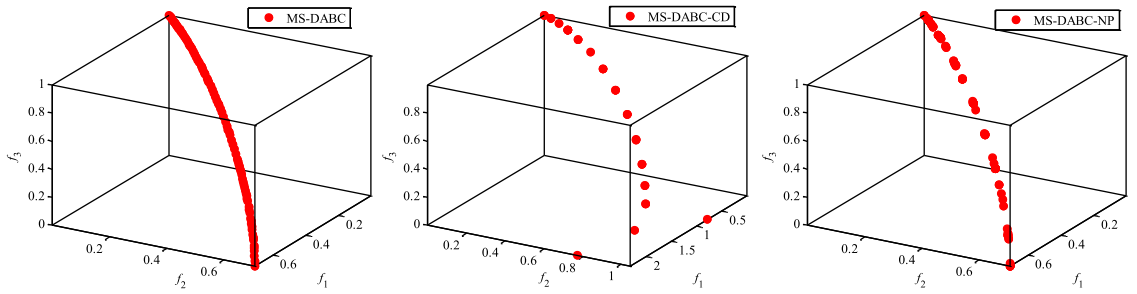

Fig. 13. Solutions for three objective DTLZ6.

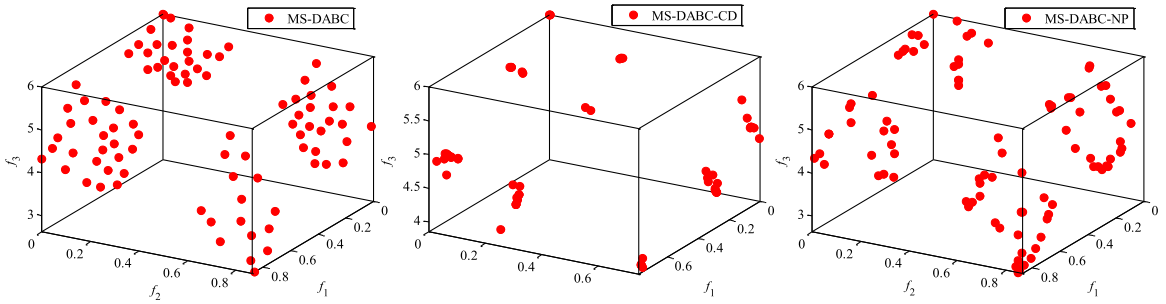

Fig. 14. Solutions for three objective DTLZ7.

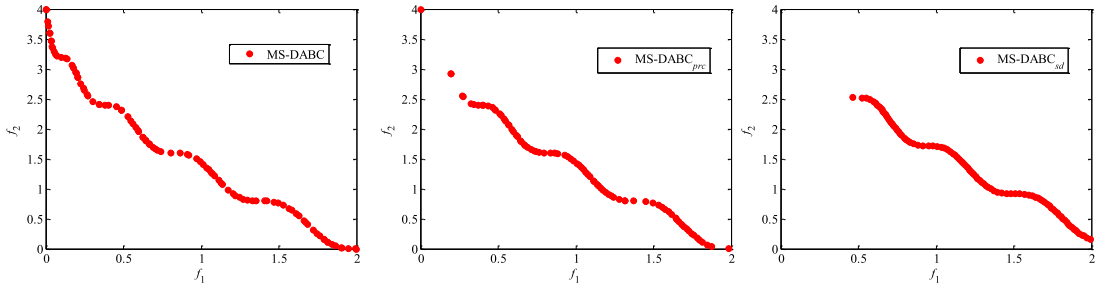

Fig. 15. Final solutions obtained by MS-DABC and its variants on the WFG1.

CD fails to cover the bottom half of the PF while MS-DABC-NP struggles to achieve uniformly distributed solutions. As for DTLZ7 with four discontinuous sub-fronts, the middle parts of four sub-fronts are not well covered for solutions obtained by MS-DABC-CD; Although MS-DABC-NP achieves a set of solutions with slightly better distribution, it still concentrates on the boundary of the front. We can therefore conclude that the better performance of MS-DABC is attributed to the elite preserving strategy for *EXA* in which diversity is given higher priority than convergence, so it is effective for the survival of the solutions with the worst front level but being critical for diversity. By contrast, the strategy in NSGA-III may result in some well diversified individuals in lower fronts being abandoned.

5.4.3. Effectiveness of the fitness assignment mechanism

In MS-DABC, the onlooker bees select the food sources to explore according to the fitness values. To assess the effects of the fitness assignment used in our algorithm, we further devise two MS-DABC variants with different fitness assignment strategies. For the first variant, the original quality indicator based fitness assignment strategy is replaced with the fitness assignment method in SPEA2, namely the MS-DABC, based on strength and density (denoted as MS-DABC$_{sd}$). That is, the fitness of a food source is determined by the strength of its dominators and density information. The second variant employs the Pareto rank value and crowding distance information for fitness calculation, namely the MS-DABC based on Pareto rank and crowding distance (denoted as MS-DABC$_{prc}$). Details of this type of fitness assignment are available in [36]. These two variants are compared with the original MS-DABC on three problems.

Figs. 15–17 visualize the final solutions obtained by MS-DABC and its two variants on three test instances. On WFG1 with the mixed and biased PF shown in Fig. 15, MS-DABC$_{sd}$ seems to struggle in approximating the true PF, while solutions obtained by MS-DABC$_{prc}$ are close to the PF, but fail to find parts of PF when f_1 approximates 0. MS-DABC obtains the best approximations with the best distribution.

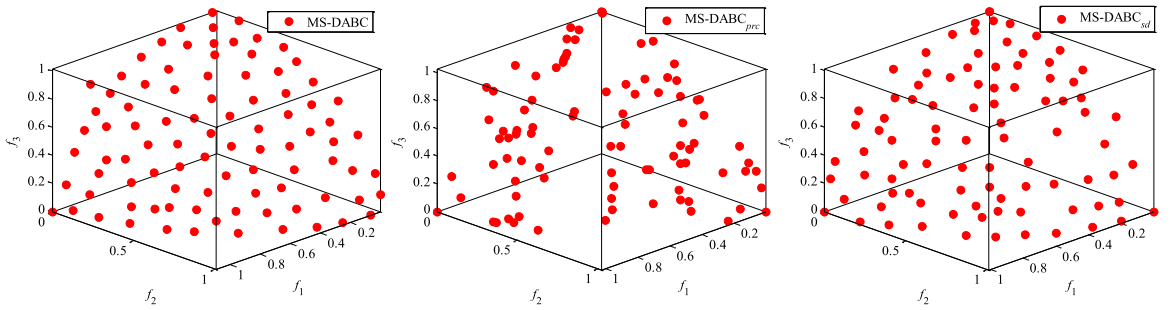

Fig. 16. Final solutions obtained by MS-DABC and its variants on the DTLZ3.

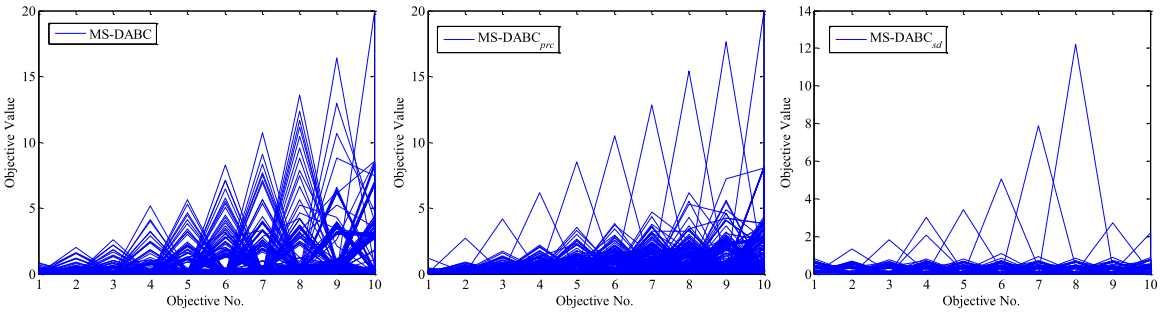

Fig. 17. The Pareto solutions obtained by MS-DABC and its variants on 10-objective WFG1 shown by the parallel coordinates.

Fig. 16 shows the final solutions found by the three algorithms on DTLZ3. Clearly, MS-DABC$_{prc}$ struggles to maintain the uniformity of the solutions, while MS-DABC and MS-DABC$_{sd}$ perform similarly and have a good solution distribution over the true PF, but the former performs slightly better than the latter in terms of uniformity.

Fig. 17 plots the final solutions obtained by the compared algorithms on the 10-objecitve WFG1, visualized by parallel coordinates. MS-DABC$_{sd}$ fails to find center solutions of the desired PF, with the solutions concentrating on the boundary points. Although a better distribution of solutions is obtained by MS-DABC$_{prc}$, there is still a clearly sparse region between the extreme points for each objective. In contrast, MS-DABC achieves relatively good approximations, although it misses some middle parts of the Pareto front in the ninth and tenth objectives. This happens mainly due to the fitness assignment mechanism in MS-DABC, which can effectively handle the dominant resistant solutions and guide the population to search towards the desired PF, especially in high dimensional objective space.

5.5. Parameter sensitivity analysis

Recall that *ng* is the sole parameter required to be tuned for achieving a good performance for our proposed algorithm. In order to investigate the impact of *ng* on MS-DABC, we carry out experiments to select the most appropriate value for *ng*. Specifically, three types of instances with different levels of difficulties (i.e., ZDT1, DTLZ2 and WFG1) are employed to study the impact of *ng* on the algorithm's ability in different environments. In our experiments, six values: 10, 20, 30, 50, 80 and 100 are considered for *ng*, and the other parameters are set as mentioned before.

The IGD values regarding the considered algorithm for the test problems over 30 runs are illustrated by the box plots in Fig. 18. Overall, the algorithm performance first improves but later deteriorates along with the increase of *ng*. Clearly, *ng* with the range [20 50] has an obvious superiority for most of the test problems in terms of IGD metric values. The box-plot for DTLZ2 implies that the *ng* set at 50 outperforms its corresponding competitors, while results from WFG1 indicate that *ng* at 20 outperforms its competitors. The performance of MS-DABC with *ng* set at 10 or 100 is usually inferior to the cases with other settings. In this paper, *ng* is set as 30 for the tradeoff.

In theory, if *ng* is small, the utilities may have little change in the given learning period, and it is hard to derive useful information from the insufficient search data so as to guide the evolution to promising regions. Additionally, the frequent re-division operations cost a lot of computational effort and affect the efficiency of the algorithm. By contrast, if *ng* is large, some subpopulations with unsuitable evolutionary operators may have a lot of individuals and evolve with a large number of generations, which results in a waste of computational effort. In addition, if the best performing evolutionary operator dominates the optimization process for a long time, the diversity of the obtained solutions may be very poor, which can induce the population to converge to some local optimal areas. From the experiments above, 30 is the most suitable tradeoff *ng* value for MS-DABC.

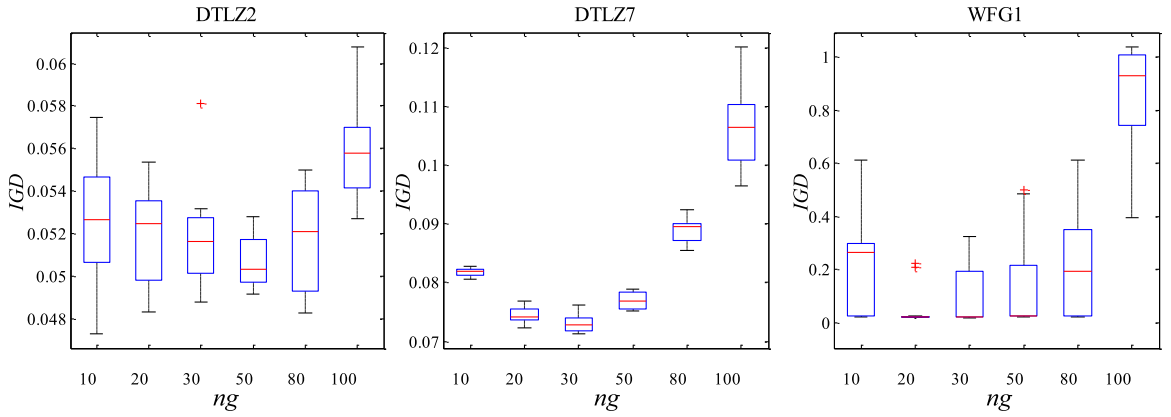

Fig. 18. Box plots of IGD metric for different values of *ng*.

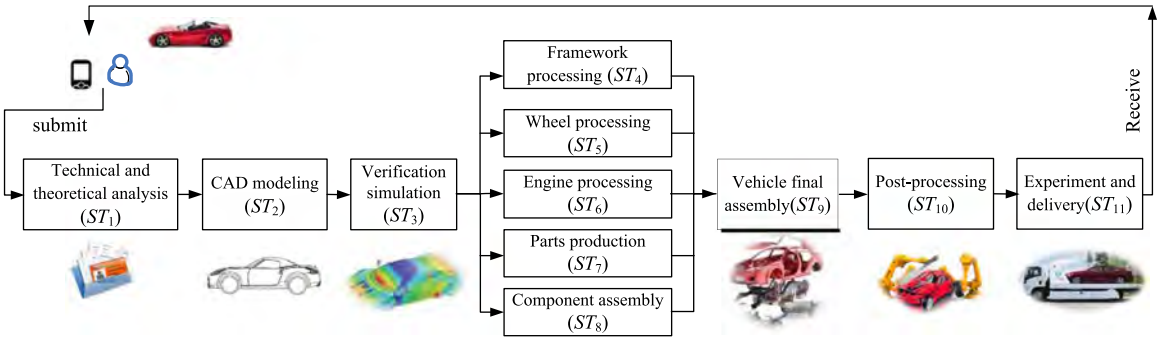

Fig. 19. The design and production of a customized car in cloud environment.

6. Case study on many-objective service composition

With the introduction of cloud technology in manufacturing, equipment and software resources can be dynamically encapsulated as cloud services. To simplify the illustration, we use "The design and production of a customized car in a cloud environment" as a case to describe the service composition process from task submission to finish. Since the manufacturing mission is complex and involves hundreds of subsystems and thousands of parts, we use a simplified model, as shown in Fig. 19. The simplified manufacturing mission is divided into several subtasks: (1) ST_1: Technical and theoretical analysis, (2) ST_2: CAD modeling, (3) ST_3: Verification simulation, (4) ST_4: Framework processing, (5) ST_5: Wheel processing, (6) ST_6: Engine processing, (7) ST_7: Parts production, (8) ST_8: Component assembly, (9) ST_9: Vehicle final assembly, (10) ST_{10}: Post-processing, and (11) ST_{11}: Experiment and delivery.

During the process, the implementation of each subtask has special requirements. For example, the CAD modeling task (ST_2) has its functionality and QoS requirements, e.g., data types, design parameters, bidding price, modeling tolerances, dimensions, etc., CMfg platform refines services with similar function properties but different in QoS, such as CATIA, UG, SolidWorks, Master-CAM or Pro/E, etc. Additionally, ST_7 can be executed by many machining services (e.g., Cutting, Stamping, Casting, Drilling, Milling, Lathing, Grindering, Screwing and Boring), albeit with different functional features and QoS (e.g., Speed, Precision, Roughness, Accuracy, Tolerance, Reliability, Maintainability, Function similarity, Credit, Time and Cost). Besides, each kind of service has its special specification for processing an object, such as material type, hardness, cutting size, tempering or cycling heat treatment. In this case, service matching is performed firstly to generate candidate service sets, and then how to choose the most appropriate from each set and to form the composite service poses a stiff challenge for an optimizer, especially when large-scale service candidates are provided.

In this section, MS-DABC is compared with several competitive algorithms (i.e., NSGA-III [5], MOEA/DD [20], MOEA/D [18], KnEA [46], SPEA/R [12]) for addressing many-objective SCOS problems, where many diverse but conflicting QoS properties need to be optimized simultaneously. The detailed MOP model for the SCOS problem is described in Section 3. In our experiments, without loss of generality, several scenarios for CMfg are considered, including 10 and 20 subtasks. In addition, five scales of the candidate service set: 20, 40, 60, 80 and 100 services for each subtask, are available for each scenario. Experiments consider the sequential structure of the composition workflow while the others can be simplified or converted into the sequential structure. Note that the Integer Array Coding Scheme is used to encode the decision vector (solution),

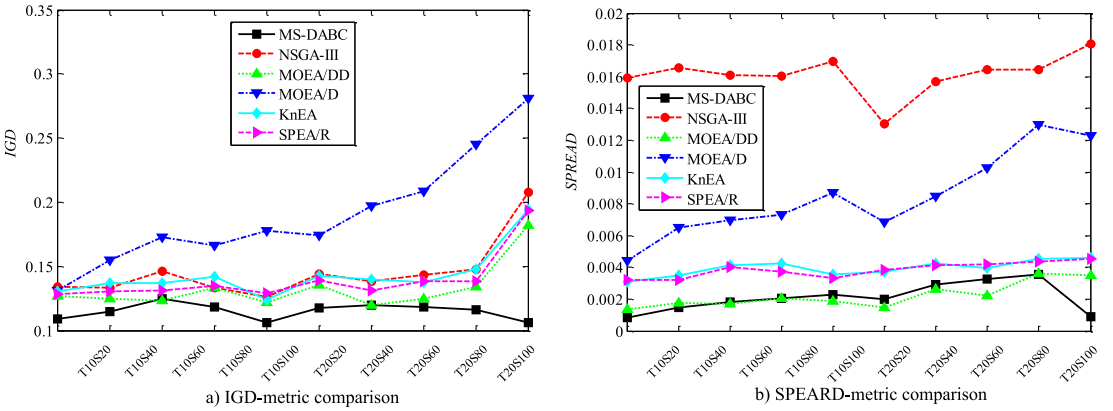

Fig. 20. Performance comparison of the algorithms on 5-objective SCOS problems.

where each position in the array represents a subtask and the value of each decision variable represents the index of a selected service for the corresponding subtask.

In many-objective contexts, two cases, i.e. 5-objective and 8-objective, are considered. For the former, five QoS-criteria have been considered, as shown in Section 3. For the latter, three more QoS parameters, Energy (*En*), Maintainability (*Ma*) and Eco-impact (*Ec*), are included. QoS values are randomly generated within the following ranges: $C \in [0, 100]$, $T \in [0, 20]$, $Rel \in [0.75, 1]$, $Aval \in [0.75, 1]$, $Rep \in [0.1, 1]$, $En \in [1, 100]$, $Ma \in [0.5, 1]$, and $Ec \in [0.6, 1]$. Then these datasets are saved in a file for uniformly testing. Besides, the specific control parameter settings of the compared algorithms are kept the same as given in the original references. The population size N is set to 210 for the 5-objective, and 156 for the 8-objective, and the $Max\_FEs = 300{,}000$. 30 independent runs for each experiment are conducted and statistical results are recorded. Since the true PFs of the SCOS problems are unknown, we independently run all the compared algorithms with 600,000 function evaluations over 50 times on each problem and take the mixed set of the non-dominated solutions (dominated and repeated solutions are discarded) as the approximate PF, which is then employed to assess the IGD metrics of the obtained solutions.

Table 12 provides the mean and standard deviation (in parentheses) of the IGD-metric values of the solutions obtained by six MOEAs on the 5- and 8-objective test instances, where the best result is marked in boldface. As can be observed, MS-DABC achieves the best IGD results in 16 out of 20 concerned instances, whereas both MOEA/DD and SPEA/R give the best results on 2 instances. The Wilcoxon's rank-sum test also indicates that MS-DABC performs significantly better than its rivals, with significant better results in 85 out of 100 comparisons. Specifically, MS-DABC outperforms NSGA-III, MOEA/DD, MOEA/D, KnEA, and SPEA/R on 19, 14, 17, 19 and 16 out of 20 instances, respectively. MOEA/DD is ranked second, followed by SPEA/R third, whereas MOEA/D and NSGA-III ranked the first and second worst, respectively. The relatively high performance of MOEA/DD and SPEA/R may be attributed to the diversity-first and convergence-second principle used in the environmental selection, where the proper use of some dominated but promising solutions is very helpful for population diversity in many-objective cases. In order to strengthen diversity, NSGA-III employs reference vectors to maintain population diversity with the help of a niche preservation operator. However, the convergence-first and diversity-second strategy may lead to some isolated but promising members being abandoned, which deteriorates the population diversity, especially when part of the PF is more difficult to search than others. In addition, reproduction without restricted mating selection is not likely to generate good offspring.

In order to observe the results more intuitively, the average IGD and SPREAD values of the solutions obtained by the compared algorithms with different problem scales are plotted in Fig. 20, from which it can be seen that the performance of MOEA/D deteriorates dramatically as the task size or the number of candidate services increase. This can be explained by the search mode "convergence outweighing diversity", which is detrimental to population diversity in a highly dimensional objective space or in complicated search landscapes. In contrast, MS-DABC shows strong versatility on SCOS with different scales. In terms of IGD, our proposed algorithm is better than or at least comparable with the other peer algorithms in all instances. As for SPREAD, the advantage of MS-DABC is not so obvious as observed in the IGD comparisons. Nevertheless, it still ranks the first or second in most instances. It seems that MS-DABC has obvious advantages over the other algorithms on larger scale problems, for example T20S100, and can lead to more diversified and well-converged individuals on the PF. The reason lies in the fact that Pareto- and decomposition-based techniques integrated in *EXA* maintenance can preserve well diversified solutions with competitive selection pressure. Furthermore, a utility function, based on the contributions of the subpopulations on *EXA* members, is employed to guide the allocation of computational resources among different subpopulations dynamically. The cooperation of these strategies improves the versatility of the proposed algorithm in tackling problems with distinct characteristics.

Table 12
IGD-metric comparisons of the algorithms for 5 and 8-objective SCOS problems.

	m	MS-DABC	NSGA-III	MOEA/DD	MOEA/D	KnEA	SPEA/R
T10S20	5	**1.09** e−1(**9.15** e−**4)1**	1.34 e−1(2.02 e−3)6+	1.27 e−1(4.08 e−3)2+	1.31 e−1(4.54 e−3)5+	1.31 e−1(4.54 e−3)4+	1.28 e−1(3.81 e−3)3+
	8	**2.21** e−**1(5.86** e−**3)1**	2.62 e−1(1.22 e−2)5+	2.52 e−1(5.30 e−3)3+	2.67 e−1(4.53 e−3)6+	2.57 e−1(8.78 e−3)4+	2.39 e−1(6.31 e−3)2+
T10S40	5	**1.15** e−**1(1.58** e−**3)1**	1.33 e−1(3.44 e−3)4+	1.25 e−1(5.39 e−3)2+	1.55 e−1(1.19 e−2)6+	1.37 e−1(5.20 e−3)5+	1.31 e−1(5.93 e−3)3+
	8	**2.24** e−**1(4.43** e−**3)1**	2.44 e−1(8.17 e−3)3+	2.44 e−1(1.04 e−2)4+	2.65 e−1(7.54 e−3)6+	2.44 e−1(5.39 e−3)2+	2.56 e−1(7.13 e−3)5+
T10S60	5	1.24 e−1(1.35 e−3)2	1.46 e−1(3.86 e−3)5+	**1.23** e−**1(1.81** e−**3)1=**	1.73 e−1(1.14 e−2)6+	1.37 e−1(4.60 e−3)4+	1.31 e−1(6.34 e−3)3+
	8	**2.46** e−**1(5.57** e−**3)1**	2.58 e−1(7.55 e−3)3+	2.52 e−1(6.85 e−3)2=	2.69 e−1(4.53 e−3)6+	2.59 e−1(9.02 e−3)4+	2.60 e−1(7.14 e−3)5+
T10S80	5	**1.18** e−**1(1.88** e−**3)1**	1.33 e−1(3.58 e−3)3+	1.33 e−1(1.13 e−2)2+	1.66 e−1(1.26 e−2)6+	1.42 e−1(8.70 e−3)5+	1.35 e−1(6.89 e−3)4+
	8	**2.48** e−**1(9.55** e−**3)1**	2.79 e−1(2.07 e−2)6+	2.68 e−1(1.21 e−2)3+	2.77 e−1(5.99 e−3)5+	2.65 e−1(1.74 e−2)2+	2.77 e−1(1.43 e−2)4+
T10S100	5	**1.06** e−**1(1.90** e−**3)1**	1.26 e−1(4.53 e−3)4+	1.22 e−1(3.19 e−3)2+	1.78 e−1(9.26 e−3)6+	1.24 e−1(2.61 e−3)3+	1.29 e−1(3.10 e−3)5+
	8	2.58 e−1(7.90 e−3)3	2.65 e−1(3.39 e−2)4=	**2.41** e−**1(3.64** e−**3)1−**	2.79 e−1(3.89 e−3)5+	2.54 e−1(6.37 e−3)2=	2.86 e−1(1.14 e−2)6+
T20S20	5	**1.17** e−**1(6.21** e−**3)1**	1.44 e−1(5.18 e−3)5+	1.36 e−1(2.29 e−2)2+	1.74 e−1(8.72 e−3)6+	1.43 e−1(8.72 e−3)4+	1.39 e−1(5.96 e−3)3+
	8	2.17 e−1(9.62 e−3)2	2.34 e−1(9.16 e−3)5+	2.30 e−1(6.58 e−3)4+	2.22 e−1(8.67 e−3)3=	2.34 e−1(9.12 e−3)6+	**1.94** e−**1(4.47** e−**3)1-**
T20S40	5	**1.19** e−**1(5.21** e−**3)1**	1.38 e−1(5.10 e−3)4+	1.20 e−1(4.30 e−3)2=	1.97 e−1(1.55 e−2)6+	1.39 e−1(7.97 e−3)5+	1.31 e−1(3.88 e−3)3+
	8	2.39 e−1(1.22 e−2)3	2.58 e−1(1.88 e−2)5+	2.53 e−1(1.98 e−2)4=	2.20 e−1(7.48 e−3)2-	2.60 e−1(1.38 e−2)6+	**2.17** e−**1(6.97** e−**3)1−**
T20S60	5	**1.18 e−1(8.47 e−3)1**	1.43 **e−**1(7.50 **e−**3)5+	1.25 **e−**1(9.88 **e−**3)2=	2.08 **e−**1(1.73 **e−**2)6+	1.37 **e−**1(9.08 **e−**3)3+	1.38 **e−**1(7.75 **e−**3)4+
	8	**2.13 e−1(9.83 e−3)1**	2.40 **e−**1(2.02 **e−**2)6+	2.38 **e−**1(1.65 **e−**2)5+	2.22 **e−**1(1.36 **e−**2)3=	2.31 **e−**1(8.29 **e−**3)4+	2.13 **e−**1(1.68 **e−**2)2=
T20S80	5	**1.16 e−1(6.41 e−3)1**	1.48 **e−**1(7.10 **e−**3)4+	1.34 **e−**1(6.89 **e−**3)2+	2.45 **e−**1(1.66 **e−**2)6+	1.48 **e−**1(6.64 **e−**3)5+	1.38 **e−**1(4.68 **e−**3)3+
	8	**2.17 e−1(1.30 e−2)1**	2.60 **e−**1(1.63 **e−**2)4+	2.64 **e−**1(1.51 **e−**2)6+	2.56 **e−**1(2.10 **e−**2)3+	2.60 **e−**1(3.09 **e−**2)5+	2.47 **e−**1(2.61 **e−**2)2+
T20S100	5	**1.06 e−1(2.03 e−3)1**	2.07 **e−**1(1.15 **e−**2)5+	1.82 **e−**1(8.68 **e−**3)2+	2.81 **e−**1(2.97 **e−**2)6+	1.94 **e−**1(1.44 **e−**2)4+	1.94 **e−**1(8.48 **e−**3)3+
	8	**2.29 e−1(1.27 e−2)1**	2.84 **e−**1(2.23 **e−**2)6+	2.60 **e−**1(2.40 **e−**2)4+	2.47 **e−**1(1.69 **e−**2)3+	2.76 **e−**1(2.17 **e−**2)5+	2.45 **e−**1(2.54 **e−**2)2=
Rank sum		26	92	55	101	82	64
Final rank		1	5	2	6	4	3
w/s/l			19/1/0	14/5/1	17/2/1	19/1/0	16/2/2

7. Conclusions and future work

Although the single-objective SCOS has been widely investigated, research on the many-objective SCOS in CMfg is still relatively scarce. In this study, a novel hybrid algorithm for many-objective SCOS is proposed by incorporating multiple size-adjustable subpopulations into the ABC framework with an external archive guided evolution mechanism. The core of the proposed hybrid algorithm, called MS-DABC, involves three aspects. First, a utility function is devised to control the allocation of computational efforts on different search operators according to the problem features or the fitness landscapes. Second, Pareto- and decomposition-based selection mechanisms are integrated into the diversity-first and convergence-second principle to achieve proper environmental selection of *EXA* such that a balance between convergence and diversity can be carefully governed, especially for many-objective problems. Third, an indicator- based fitness assignment scheme, depending on a solution's contribution in HV, is employed to identify potential regions to be exploited by onlookers. Comprehensive experiments on benchmark test problems have been carried out to verify the effectiveness and competitiveness of the proposed algorithm.

This study focuses on many-objective SCOS problems solved by the proposed MS-DABC, which will be extended to handle SCOS in more practical application contexts (e.g., correlation-aware situations and multi-task dynamical scenarios) in our future work so that it can promote intelligent decision making of CMfg platform. Further, the operator configuration and resource dynamical allocation is very useful in computationally expensive optimization problems, where there is still room for exploring other approaches for adjusting computational efforts in different contexts. This also will be considered in our future work.

Acknowledgments

The project was supported by the National Natural Science Foundation of China under Grant nos. 51675186 and 51175187, the Science & Technology Foundation of Guangdong Province under Grant no. 2017A030223002. The first author wishes to acknowledge the financial support of the China Scholarship Council (CSC) and the Excellent Doctoral Dissertation Innovation Fund of South China University of Technology (SCUT).

References

[1] J. Bader, E. Zitzler, HypE: an algorithm for fast hypervolume-based many-objective optimization, Evol. Comput. 19 (2008) 45–76.
[2] N. Beume, B. Naujoks, M. Emmerich, SMS-EMOA: multiobjective selection based on dominated hypervolume, Eur. J. Oper. Res. 181 (2007) 1653–1669.
[3] J. Brest, S. Greiner, B. Boskovic, M. Mernik, V. Zumer, Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems, IEEE Trans. Evol. Comput. 10 (2006) 646–657.
[4] J.X. Cheng, G.G. Yen, G.X. Zhang, A many-objective evolutionary algorithm with enhanced mating and environmental selections, IEEE Trans. Evol. Comput. 19 (2015) 592–605.
[5] K. Deb, H. Jain, An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach. Part I: solving problems with box constraints, IEEE Trans. Evol. Comput. 18 (2014) 577–601.
[6] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comput. 6 (2002) 182–197.
[7] R.H.a. Gómez, C.A.C. Coello, Improved metaheuristic based on the R2 indicator for many-objective optimization, in: Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation, ACM, Madrid, Spain, 2015, pp. 679–686.
[8] D. Gong, F. Sun, J. Sun, X. Sun, Set-based many-objective optimization guided by a preferred region, Neurocomputing 228 (2017) 241–255.
[9] W. He, L. Xu, A state-of-the-art survey of cloud manufacturing, Int. J. Comput. Integr. Manuf. 28 (2015) 239–250.
[10] Z. He, G.G. Yen, J. Zhang, Fuzzy-based pareto optimality for many-objective evolutionary algorithms, IEEE Trans. Evol. Comput. 18 (2014) 269–285.
[11] B. Huang, C. Li, F. Tao, A chaos control optimal algorithm for QoS-based service composition selection in cloud manufacturing system, Enterp. Inf. Syst. 8 (2014) 445–463.
[12] S.Y. Jiang, S.X. Yang, A strength pareto evolutionary algorithm based on reference direction for multiobjective and many-objective optimization, IEEE Trans. Evol. Comput. 21 (2017) 329–346.
[13] A. Jula, E. Sundararajan, Z. Othman, Cloud computing service composition: a systematic literature review, Expert Syst. Appl. 41 (2014) 3809–3824.
[14] D. Karaboga, B. Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, J. Global Optim. 39 (2007) 459–471.
[15] D. Karaboga, B. Gorkemli, C. Ozturk, N. Karaboga, A comprehensive survey: artificial bee colony (ABC) algorithm and applications, Artif. Intell. Rev. 42 (2014) 21–57.
[16] Y. Laili, F. Tao, L. Zhang, Y. Cheng, Y. Luo, B.R. Sarker, A ranking chaos algorithm for dual scheduling of cloud service and computing resource in private cloud, Comput. Ind. 64 (2013) 448–463.
[17] B.H. Li, L. Zhang, S.L. Wang, F. Tao, J.W. Cao, X.D. Jiang, X. Song, X.D. Chai, Cloud manufacturing: a new service-oriented networked manufacturing model, Comput. Integr. Manuf. Syst. 16 (2010) 1–16.
[18] H. Li, Q.F. Zhang, Multiobjective optimization problems with complicated pareto sets, MOEA/D and NSGA-II, IEEE Trans. Evol. Comput. 13 (2009) 284–302.
[19] K. Li, K. Deb, Q. Zhang, Q. Zhang, Efficient nondomination level update method for steady-state evolutionary multiobjective optimization, IEEE Trans. Cybern. 47 (2017) 2838–2849.
[20] K. Li, K. Deb, Q.F. Zhang, S. Kwong, An evolutionary many-objective optimization algorithm based on dominance and decomposition, IEEE Trans. Evol. Comput. 19 (2015) 694–716.
[21] K. Li, A. Fialho, S. Kwong, Q.F. Zhang, Adaptive operator selection with bandits for a multiobjective evolutionary algorithm based on decomposition, IEEE Trans. Evol. Comput. 18 (2014) 114–130.
[22] K. Li, S. Kwong, K. Deb, A dual-population paradigm for evolutionary multiobjective optimization, Inf. Sci. 309 (2015) 50–72.
[23] M. Li, S. Yang, X. Liu, Shift-based density estimation for pareto-based algorithms in many-objective optimization, IEEE Trans. Evol. Comput. 18 (2014) 348–365.
[24] H.L. Liu, L. Chen, Q. Zhang, K. Deb, Adaptively allocating search effort in challenging many-objective optimization problems, IEEE Trans. Evol. Comput. (2017) in press, doi:10.1109/TEVC.2017.2725902.
[25] Y. Liu, D. Gong, J. Sun, Y. Jin, A many-objective evolutionary algorithm using a one-by-one selection strategy, IEEE Trans. Cybern. 47 (2017) 2689–2702.
[26] Y. Liu, D. Gong, X. Sun, Y. Zhang, Many-objective evolutionary optimization based on reference points, Appl. Soft Comput. 50 (2017) 344–355.

[27] R. Mallipeddi, P.N. Suganthan, Q.K. Pan, M.F. Tasgetiren, Differential evolution algorithm with ensemble of parameters and mutation strategies, Appl. Soft Comput. 11 (2011) 1679–1696.
[28] A.J. Nebro, J.J. Durillo, G. Nieto, C.A.C. Coello, F. Luna, E. Alba, SMPSO: a new pso-based metaheuristic for multi-objective optimization, in: Proceedings of the 2009 IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making, Nashville, TN, USA, 2009, pp. 66–73.
[29] A.K. Qin, V.L. Huang, P.N. Suganthan, Differential evolution algorithm with strategy adaptation for global numerical optimization, IEEE Trans. Evol. Comput. 13 (2009) 398–417.
[30] Q.H. Tang, Z.X. Li, L.P. Zhang, An effective discrete artificial bee colony algorithm with idle time reduction techniques for two-sided assembly line balancing problem of type-II, Comput. Ind. Eng. 97 (2016) 146–156.
[31] F. Tao, Y. Feng, L. Zhang, T.W. Liao, CLPS-GA: a case library and Pareto solution-based hybrid genetic algorithm for energy-aware cloud service scheduling, Appl. Soft Comput. 19 (2014) 264–279.
[32] F. Tao, Y. LaiLi, L. Xu, L. Zhang, FC-PACO-RM: a parallel method for service composition optimal-selection in cloud manufacturing system, IEEE Trans. Ind. Inf. 9 (2013) 2023–2033.
[33] F. Tao, K. Qiao, L. Zhang, Z. Li, A.Y.C. Nee, GA-BHTR: an improved genetic algorithm for partner selection in virtual manufacturing, Int. J. Prod. Res. 50 (2012) 2079–2100.
[34] F. Tao, D. Zhao, Y. Hu, Z. Zhou, Correlation-aware resource service composition and optimal-selection in manufacturing grid, Eur. J. Oper. Res. 201 (2010) 129–143.
[35] Y. Tian, R. Cheng, X. Zhang, F. Cheng, Y. Jin, An indicator based multi-objective evolutionary algorithm with reference point adaptation for better versatility, IEEE Trans. Evol. Comput. (2017) in press, doi:10.1109/TEVC.2017.2749619.
[36] D.H. Tran, M.Y. Cheng, M.T. Cao, Hybrid multiple objective artificial bee colony with differential evolution for the time-cost-quality tradeoff problem, Knowl.-Based Syst. 74 (2015) 176–186.
[37] H. Wang, L. Jiao, X. Yao, Two_Arch2: an improved two-archive algorithm for many-objective optimization, IEEE Trans. Evol. Comput. 19 (2015) 524–541.
[38] G.H. Wu, R. Mallipeddi, P.N. Suganthan, R. Wang, H.K. Chen, Differential evolution with multi-population based ensemble of mutation strategies, Inf. Sci. 329 (2016) 329–345.
[39] Y. Xiang, Y. Zhou, M. Li, Z. Chen, A vector angle-based evolutionary algorithm for unconstrained many-objective optimization, IEEE Trans. Evol. Comput. 21 (2017) 131–152.
[40] S. Yang, M. Li, X. Liu, J. Zheng, A grid-based evolutionary algorithm for many-objective optimization, IEEE Trans. Evol. Comput. 17 (2013) 721–736.
[41] Y. Yuan, H. Xu, B. Wang, X. Yao, A new dominance relation-based evolutionary algorithm for many-objective optimization, IEEE Trans. Evol. Comput. 20 (2016) 16–37.
[42] Y. Yuan, H. Xu, B. Wang, B. Zhang, X. Yao, Balancing convergence and diversity in decomposition-based many-objective optimizers, IEEE Trans. Evol. Comput. 20 (2016) 180–198.
[43] J.Q. Zhang, A.C. Sanderson, JADE: adaptive differential evolution with optional external archive, IEEE Trans. Evol. Comput. 13 (2009) 945–958.
[44] Q. Zhang, W. Liu, H. Li, The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances, in: Proceedings of the 2009 IEEE Congress on Evolutionary Computation, 2009, pp. 203–208.
[45] Q.F. Zhang, H. Li, MOEA/D: a multiobjective evolutionary algorithm based on decomposition, IEEE Trans. Evol. Comput. 11 (2007) 712–731.
[46] X.Y. Zhang, Y. Tian, Y.C. Jin, A knee point-driven evolutionary algorithm for many-objective optimization, IEEE Trans. Evol. Comput. 19 (2015) 761–776.
[47] S.Z. Zhao, P.N. Suganthan, Q.F. Zhang, Decomposition-based multiobjective evolutionary algorithm with an ensemble of neighborhood sizes, IEEE Trans. Evol. Comput. 16 (2012) 442–446.
[48] J. Zhou, X. Yao, A hybrid approach combining modified artificial bee colony and cuckoo search algorithms for multi-objective cloud manufacturing service composition, Int. J. Prod. Res. 55 (2017) 4765–4784.
[49] J. Zhou, X. Yao, A hybrid artificial bee colony algorithm for optimal selection of QoS-based cloud manufacturing service composition, Int. J. Adv. Manuf. Technol. 88 (2017) 3371–3387.
[50] E. Zitzler, M. Laumanns, L. Thiele, SPEA2: improving the strength pareto evolutionary algorithm for multiobjective optimization, in: Proceedings of Evolutionary Proceedings of the Eurogen'2001, Athens. Greece, 2001.