

Received January 9, 2020, accepted January 27, 2020, date of publication February 13, 2020, date of current version March 11, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2973670

# A Survey of Multiobjective Evolutionary Algorithms Based on Decomposition: Variants, Challenges and Future Directions

QIAN XU<sup>1</sup>, ZHANQI XU<sup>1</sup>, (Senior Member, IEEE), AND TAO MA<sup>1</sup>, (Member, IEEE)

State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an 710071, China

Corresponding author: Zhanqi Xu (zqxu@mail.xidian.edu.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2019YFB1803605, and in part by the National Natural Science Foundation of China under Grant 61572391.

**ABSTRACT** There are many challengeable multiobjective optimization problems in different areas, whose optimization objectives are usually diversionary. Decomposition methods and evolution mechanisms enable multiobjective evolutionary algorithms based on decomposition (MOEA/D) to tackle these complex optimization problems efficiently. Therefore, MOEA/D has found wide applications in various fields and been attracting increasingly significant attention from both academia and industry since it was first proposed by Zhang and Li in 2007. Many efforts that are dedicated to improving and extending MOEA/D have been summarized shortly by some papers in their introductions, and there exists only one article that reviewed MOEA/D comprehensively in 2017. However, a number of MOEA/D variants with novel methods solving versatile problems in different fields have been emerging since then. This article is motivated by a more systematic survey of MOEA/D from its original ideas to edge-cutting works, including its basic framework and a comprehensive overview of the improvements on key components (decomposition method, weight vector generation method, and evolutionary operator) and the extensions to both many-objective and constrained multiobjective optimizations. The findings of this survey are categorized in seven aspects with corresponding references. In addition, different from introducing briefly the future research directions of MOEA/D in conclusion of the survey in 2017, we present a more detailed outlook that explores not only the novel challenges but also the future research directions, including three aspects in theory and application researches, its challenges in many-objective optimization, and some issues applying MOEA/D to the cutting-edge areas. It is expected that our work will help researchers to start their MOEA/D-based investigations.

**INDEX TERMS** Multiobjective evolutionary algorithms based on decomposition (MOEA/D), decomposition method, weight vector generation method, evolutionary operator, many-objective optimization.

## I. INTRODUCTION

There are many multiobjective optimization problems (MOPs) in various fields, for example, how to reasonably allocate resources in the network for implementing several goals jointly, such as maximizing resource utilization and minimizing operational expenditure [1]. These optimization objectives usually conflict with each other, in such a way that an objective cannot be improved without causing degradation to some others. Therefore, it is impossible to make them all

optimal at the same time, and we could seek multiple Pareto optimal solutions (POs) for a tradeoff among all objectives instead of a globally optimal solution. For any two feasible solutions  $x_1$  and  $x_2$  of an MOP,  $x_1$  dominates  $x_2$  if  $x_1$  is not inferior to  $x_2$  on all objectives and  $x_1$  is superior to  $x_2$  on at least one objective, and  $x_1$  is a PO if it is not dominated by all the others in the solution set.

Single-objective optimization algorithms that often find a single optimal solution are no longer suitable for solving MOPs. Some researchers have proposed scalarization-based techniques and meta-heuristic algorithms based on swarm intelligence, and corresponding examples are the weighted

The associate editor coordinating the review of this manuscript and approving it for publication was Sotirios Goudos<sup>1</sup>.

sum and evolutionary algorithms (EAs). The multiobjective evolutionary algorithms (MOEAs) based on multi-population co-evolution (CE) have gained great popularity in recent years. In a single simulation run, a set of POSs with optimal tradeoffs among approximately conflicting objective functions can be obtained, and decision makers (DMs) can thus choose solutions based on different preferences.

According to different selection strategies, MOEAs can be broadly classified into three categories that are based on domination, indicator and decomposition [2]. When the number of objectives becomes large, it is increasingly hard for domination-based MOEAs to select solutions, because almost all the solutions in the population become nondominated with each other. In the second category, the exact indicator calculation is highly complex and the computational resources required grow exponentially with the increasing number of objectives. Compared with other MOEAs, the decomposition-based framework can enhance population diversity by using a group of weight vectors or reference points distributed properly, and allow parallel computation to speed up the solution convergence. As a representative of decomposition-based MOEAs, MOEA/D has been receiving focus on the multiobjective optimization field since it was introduced in 2007 [3]. It decomposes an MOP into a number of single-objective subproblems through aggregation functions and simultaneously optimizes them by using EAs. Each individual in the population represents the current best solution to one of the subproblems, and each subproblem is optimally solved with the solutions of all its surrounding subproblems.

Thanks to the inherent advantages in terms of decomposition and parallelism, MOEA/D has been widely used to solve many practical engineering optimization problems, including: 1) machine learning problems, most of which can be attributed to optimization problems, and sometimes have more than one objective to optimize, such as fuzzy classifier design [4], optimization of deep neural network connection structure [5] and determination of a hyper-parameter for the regularization term of convolutional neural network [6]; 2) scheduling problems, allocating resources to different tasks in real time under certain constraints, such as hybrid flow shop scheduling [7], job shop scheduling [8], and order scheduling [9]; 3) design problems, usually achieving the goals of low cost, low consumption, and large profit while meeting the needs of users, such as wireless sensor network coverage design [10] and resource optimization for network function virtualization (NFV) requests [11]; 4) control problems, addressing the optimal parameters and mechanism in such a system as the Internet of Things [12] or reservoir flood control [13]; 5) operation problems, studying the cost reduction in the operation and maintenance of a system like hybrid energy systems [14]; and 6) investment problems, considering how to invest limited funds in a number of projects to maximize returns like portfolio optimization problem [15].

This article consists mainly of three parts: 1) a comprehensive overview of the improvements and the extensions

with some new techniques; 2) findings of this survey in seven aspects; and 3) some challenges and research directions in four categories. Our work makes three contributions to the field of multiobjective optimization: 1) different from the earlier survey [2], this is a more comprehensive and timely survey covering the recent advanced studies on MOEA/D, which provides innovative research ideas and methods for scholars. Further, we organize the article according to the methods or techniques used by these variants in the review, which will help readers to find appropriate methods to solve the practical problems; 2) findings of this survey in seven aspects are discussed, including the utilized techniques and related references. The tables and figures in this article will aid researchers to understand the algorithms more systematically and to select some algorithms they are interested in for further study; 3) new challenges and future research directions are discussed from diverse perspectives, such as theoretical analysis, practical application, and hotspot topics, which are more detailed and timely compared to the survey of MOEA/D in 2017. We believe that this survey will attract more researchers to pay attention to MOEA/D and encourage them to develop more efficient and intelligent algorithms, so as to cope with new challenges in solving complex optimization problems, especially in some cutting-edge areas of large-scale optimization, machine learning, software defined networking (SDN) and NFV.

The rest of this article is organized as shown in Fig. 1. Section II describes the framework and characteristics of MOEA/D, and outlines its different research directions. Section III discusses the detailed component improvements of MOEA/D and its extensions to other fields. Section IV presents the findings of this survey. Open problems or challenges and possible solutions are addressed in Section V. The conclusion is given in Section VI. For convenience, the list of abbreviations used in this article is presented in Table 1.

## II. MOEA/D OVERVIEW

### A. MOEA/D FRAMEWORK

Decomposition is the main point in the MOEA/D framework, so an MOP with  $m$  objective functions could be converted into  $N$  subproblems that are represented by  $N$  weight vectors. The population size  $N$  (the number of subproblems or weight vectors), the neighborhood size  $T$ , a set of weight vectors  $(\lambda^1, \dots, \lambda^N)$  distributed uniformly, and the maximum number of generations  $G$  are its inputs. We present the pseudocode and main workflow of MOEA/D in Algorithm 1 and Fig. 2, respectively. The original MOEA/D mainly includes the following three steps.

#### 1) INITIALIZATION OF EXTERNAL POPULATION, NEIGHBORHOOD, POPULATION, AND REFERENCE POINT

The initialization process consists of the following four actions. External population (EP) stores the nondominated

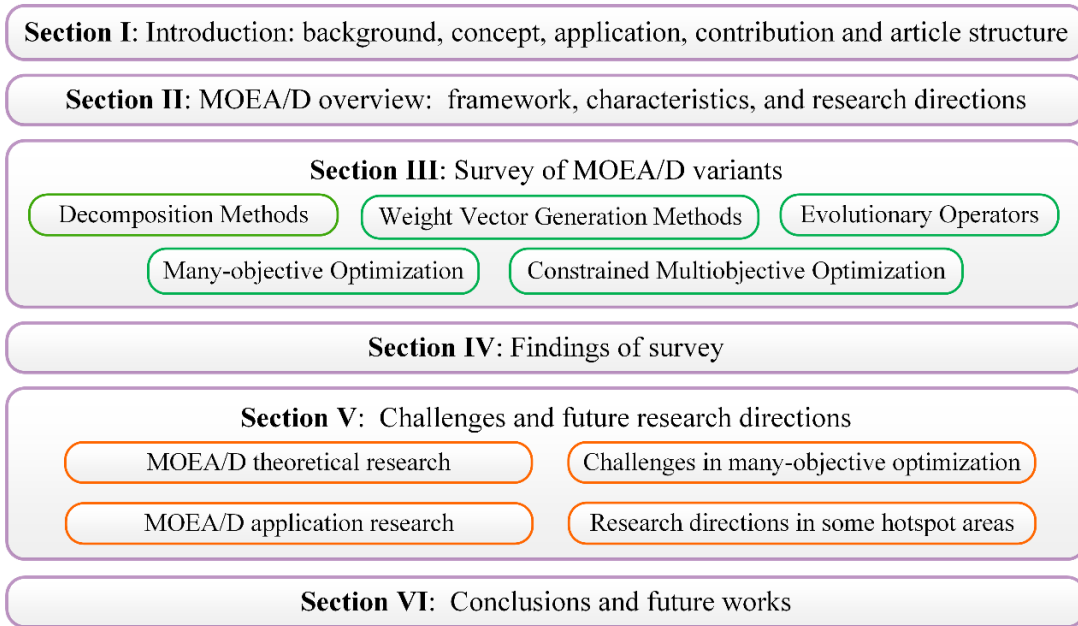


FIGURE 1. Organization of article.

solutions found during a search, which is actually empty when the algorithm starts running. The neighborhood of a specific subproblem is the set of  $T$  subproblems closest to it, which can be obtained by calculating the Euclidean distance between its weight vector and the weight vector of any other subproblem. The initial population can be generated randomly and the objective function value  $F(x^i) = (f_1(x^i), \dots, f_m(x^i))^T$  for each individual  $x^i$  is calculated accordingly. The reference point,  $z = (z_1, \dots, z_m)^T$ , denotes the vector that is composed of the best objective values, where  $z_i$  represents the best value obtained so far for the  $i$ th-objective.

### 2) EVOLUTION OPERATION FOR EACH SUBPROBLEM

Parents are selected with mating selection techniques from the neighborhood of the current subproblem, and a new solution  $y$  is generated by applying reproduction operators. If  $y$  is infeasible, a problem-specific repair algorithm would be applied on  $y$  to produce a solution  $y'$ . Then we perform the replacement operation. Especially, we update the reference point by comparing the fitness values between the previous point and  $y'$ , and the one with a better fitness value is retained. The update process for neighboring solutions is similar to that for the reference point. For EP update at the end of each generation, we remove all the vectors dominated by  $F(y')$  from EP and add  $F(y')$  to EP if no vectors in EP dominate it. Thus, the resulting EP consists of the best solutions that have been found for  $N$  subproblems so far.

### 3) JUDGMENT OF THE ALGORITHM TERMINATION

If the number of iterations reaches  $G$ , then the algorithm stops and outputs EP, otherwise goes to step 2. The evolution result is to obtain the Pareto set (PS) and the Pareto front (PF),

which refer to the set of all the Pareto optimal solution points in the decision space and the set of all the Pareto optimal objective vectors in the objective space, respectively [2].

### B. CHARACTERISTICS OF MOEA/D

MOEA/D combines the traditional mathematical decomposition method with EAs, making it easier to apply the single-objective optimizer to each subproblem associated with a solution, thus maintaining the solution dispersed. In addition, the neighborhood structure introduced in MOEA/D allows each subproblem to be optimized by using the information only from its neighboring subproblems, greatly accelerating the algorithm convergence. Therefore, compared with traditional MOEAs without decomposition, the MOEA/D framework can significantly reduce the difficulty in both fitness allocation and diversity control [3].

Although MOEA/D has many advantages, it has the following limitations: 1) fixed weight vectors are not suitable for MOPs with complex PF shapes; 2) it is not easy to set a reasonable neighborhood size because a larger size wastes computational resources while a smaller size easily makes the search fall into local optimum; 3) MOEA/D faces many difficulties in solving many-objective optimization problems (MaOPs), which refer to the optimization problems with four or more objectives; 4) evolutionary operators need to be improved to accommodate different problems. Therefore, many methods have been investigated to improve the above deficiencies.

### C. RESEARCH DIRECTIONS OF MOEA/D

As shown in Fig. 3, the research on MOEA/D mainly involves two aspects of component improvements and extensions to

TABLE 1. List of abbreviations.

ACD	adaptively constrained decomposition	MOP	multiobjective optimization problem
ACDP	angle-based constrained dominance principle	MSF	multiplicative scalarizing function
ACO	ant colony optimization	NBI	normal boundary intersection
AI	artificial intelligence	NFV	network function virtualization
APS	adaptive penalty scheme	NUMS	nonuniform mapping scheme
AWA	adaptive weight vector adjustment	OBL	opposition-based learning
Capex	capital expenditures	Opex	operating expenses
CDAS	control of dominance area of solutions	PaP	Pareto adaptive PBI
CDG	constrained decomposition with grids	$p\lambda$	Pareto-adaptive weight vectors
CDP	constrained dominance principle	PBI	penalty-based boundary intersection
CE	co-evolution	PF	Pareto front
CMA-ES	covariance matrix adaptation evolution strategy	pMOEA/D	MOEA/D with biased weight vector
CMOP	constrained multiobjective optimization problem	POS	Pareto optimal solution
cwMOEA/D	MOEA/D with the co-evolution of weights	PPS	push and pull search
DE	differential evolution	PS	Pareto set
DM	decision maker	PSF	penalty-based scalarizing function
DrEA	D-dominance relationship-based evolutionary algorithm	$p$ -TCH	Tchebycheff decomposition with $L_p$ norm constraint
EA	evolutionary algorithm	RA-NFV	resource allocation in NFV
EONs	elastic optical networks	R-MEAD	reference point based multiobjective evolutionary algorithm with decomposition
EP	external population	ROI	region of interest
GLP	good lattice point	RVEA	reference vector guided EA
GR	global replacement	SDN	software defined networking
IPBI	inverted PBI	SFC	service function chain
$I_{SDE}$	indicator based on the shift-based density estimation	SOM	self-organizing map
LTD	learning-to-decomposition	SPS	subproblem-based penalty scheme
M2M	multiple to multiple	SR	stochastic ranking
MaOP	many-objective optimization problem	TCH	Tchebycheff
MOEA	multiobjective evolutionary algorithm	T-MOEA/D	MOEA/D with target region
MOEA/D	MOEA based on decomposition	UDEA/D	decomposition-based EA with uniform design
MOEA/DD	MOEA/D with domination archive	UDM	uniform decomposition measurement
MOEA/D-DI	MOEA/D with diversity improvement	UMOEA/D	uniform design MOEA/D
MOEA/D-DU	MOEA/D with a distance-based updating	VNF	virtual network function
MOEA/D-RW	MOEA/D with fixed and random weight vectors	WAD	weight vector adaptation
MOEA/D-UD	MOEA/D with uniform design	WM	weighted mixture-style
MOEA/HD	MOEA based on hierarchical decomposition	WS	weighted sum

other fields. The motivation for improvement is to overcome the limitations in designing components of the MOEA/D framework and to improve algorithm performance for solving various complex problems. For the wider application, MOEA/D needs to be further extended to some increasingly concerned areas, such as many-objective optimization and constrained multiobjective optimization.

The main improvement on the original MOEA/D components could be divided into three subspects further, including

decomposition methods, weight vector generation methods, and evolutionary operators. Traditional decomposition methods have been improved [16]–[19], combined with each other [20], [22] and adaptively selected [23], [24]. Liu *et al.* [25] decomposed an MOP into several multiobjective subproblems without using any aggregation functions. Weight vectors constructed by the simplex lattice design [3] cannot guarantee the uniformity of the solution set on some issues. Researchers proposed various techniques to improve the

**Algorithm 1** MOEA/D

---

**Input:**  $N$ : population size  
 $\lambda^1, \dots, \lambda^N$ : weight vectors distributed uniformly  
 $T$ : neighborhood size  
 $G$ : maximum generation

**Output:**  $EP$ : external population

**{Step I: Initialization}**

- 1 set  $EP = \emptyset$ ;
- 2 **foreach**  $i \neq j \leq N$  **do**
- 3 | calculate Euclidean distance between  $\lambda^i$  and  $\lambda^j$ ;
- 4 **end**
- 5 **foreach**  $i \leq N$  **do**
- 6 | set neighborhood of  $i$  as  $B(i) = \{i_1, \dots, i_T\}$ , where  $\lambda^{i_1}, \dots, \lambda^{i_T}$  are the  $T$  closest weight vectors to  $\lambda^i$ ;
- 7 **end**
- 8 generate initial population  $\{x^1, \dots, x^N\}$  randomly;
- 9 calculate objective function  $F(x^i) = (f_1(x^i), \dots, f_m(x^i))^T$  of each individual  $x^i$  in the initial population;
- 10 initialize  $z = (z_1, \dots, z_m)^T$ , where  $z_i = \min\{f_i(x^1), \dots, f_i(x^N)\}$ ;

**{Step II: Evolution}**

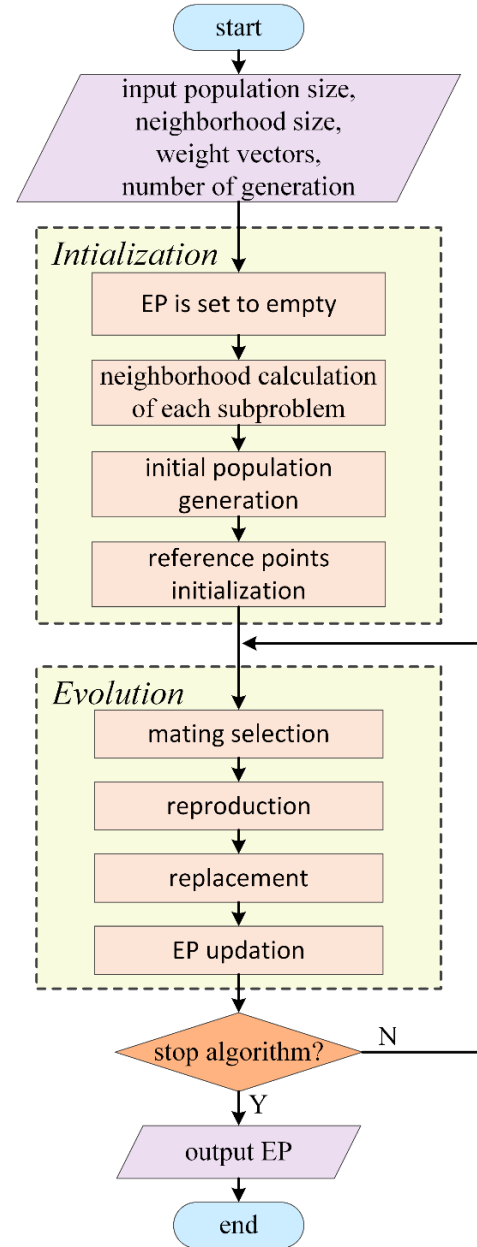
- 11 **foreach**  $i \leq N$  **do**
- 12 | **mating selection:** select two index  $k$  and  $l$  from  $B(i)$ ;
- 13 | **reproduction:** get a new solution  $y$  from  $x^k$  and  $x^l$  by applying reproduction operators;
- 14 | apply a problem-specific repair algorithm on  $y$  to produce  $y'$ ;
- 15 | **foreach**  $j = 1, \dots, m$  **do**
- 16 | | **if**  $f_j(y') < z_j$  **then**
- 17 | | | set  $f_j(y') = z_j$ ;
- 18 | | **end**
- 19 | **end**
- 20 | **foreach**  $j \in B(i)$  **do**
- 21 | | **if**  $g^{dec}(y'|\lambda^j, z) \leq g^{dec}(x^j|\lambda^j, z)$  **then**
- 22 | | | set  $x^j = y'$  and  $F(x^j) = F(y')$ ;
- 23 | | **end**
- 24 | **end**
- 25 | delete all vectors dominated by  $F(y')$  from  $EP$ ;
- 26 | **if** no vectors in  $EP$  dominate  $F(y')$  **then**
- 27 | | add  $F(y')$  to  $EP$ ;
- 28 | **end**
- 29 **end**

**{Step III: Stopping Switch}**

- 30 **if** iterations =  $G$  **then**
- 31 | output  $EP$ ;
- 32 **else**
- 33 | go to **Step II**;
- 34 **end**

---

weight vector generation method, such as uniform design [26], dynamic adjustment based on computational resources



**FIGURE 2.** MOEA/D flowchart.

required by each subproblem [27] or PF geometry shape [28], [29], user preferences-based [30], and hybrid [31]. Evolution operators consist of mating selection, reproduction and replacement. The range and related techniques of mating selection [32]–[34] and replacement mechanisms [35], [36] have been investigated to enhance population diversity. New reproduction operators like differential evolution (DE) [32], [37] and other heuristic algorithms [38] have been introduced into MOEA/D. The other investigations addressed the extension of the MOEA/D framework to many-objective optimization [39]–[47], constrained multiobjective optimization [48]–[57], and real-world optimization [4]–[15]. For the five aspects shown in Fig. 3, we will present a detailed review of each aspect in the section below.

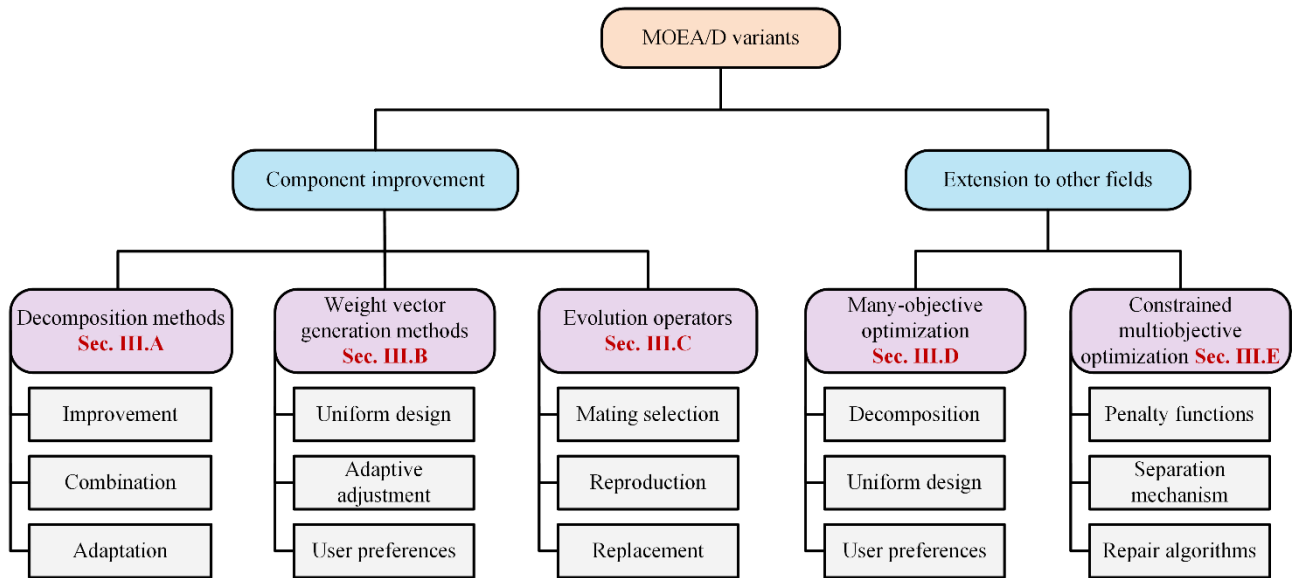


FIGURE 3. Research on MOEA/D variants.

### III. SURVEY OF MOEA/D VARIANTS

This section surveys the MOEA/D variants in two aspects. The first focuses on the improvements over key components of the original MOEA/D, including decomposition methods, weight vector generation methods, and evolutionary operators. These improvements mainly investigate the techniques of uniform design, combination, adaptation, and user preferences-based. The second discusses the studies about extending the MOEA/D framework to many-objective optimization and constrained multiobjective optimization. The former addresses the extension difficulty and methods that are integrated into MOEA/D to overcome these challenges, and the latter briefly summarizes several constraint handling techniques used widely for dealing with infeasible solutions. Due to limited space, we just briefly describe the application of MOEA/D to real-world optimization problems in Section I.

#### A. DECOMPOSITION METHODS

Decomposition methods play a fundamental role in the performance of MOEA/D and its variants. MOPs cannot be well converted into appropriate subproblems if the decomposition method is selected unreasonably for a specific problem, then MOEA/D may fail to approximate PF. Several typical decomposition methods have been investigated in the original MOEA/D, such as the weighted Tchebycheff (TCH), the weighted sum (WS), and the penalty-based boundary intersection (PBI). However, these methods with fixed parameters result in solving MOPs with different PF shapes inappropriately. Moreover, due to the properties of decomposition method contours (a set of equal scalarizing function values), the improvement region corresponding to the method

becomes so large that population diversity declines in some problems. As shown in the following four major aspects, some efforts have been made on the typical methods to address these deficiencies and the challenge of determining appropriate decomposition methods for different optimization problems.

#### 1) IMPROVEMENT OF THE ORIGINAL DECOMPOSITION METHODS

The original PBI decomposition method becomes unreliable if the obtained reference point is far from the ideal point, then we could not get the expected PF. An inverted PBI (IPBI) decomposition method was presented in [16], which is an extension of conventional PBI and WS. PBI evolves solutions toward the reference point by minimizing the scalarizing function value. Contrarily, IPBI evolves solutions from the worst objective vector in the current population by maximizing the scalarizing function value, i.e., a solution with a larger  $d_1$  is considered a better solution close to the PF, where  $d_1$  denotes the distance between the reference point and the projection of Pareto optimal (objective) vector in the weight vector direction. IPBI can conduct a wide range of searches on the objective space that WS cannot reach, thus enabling the PF to be disseminated widely.

Yang et al. [60] studied the impact of the penalty factor in PBI on algorithm performance, which is not considered in [16]. The results concluded that a smaller factor is needed for a faster convergence during the early search, while a larger factor is required to maintain population diversity at the later search stage. Then, two new penalty schemes, adaptive penalty scheme (APS) and subproblem-based penalty scheme

(SPS), were suggested to adaptively adjust the penalty factor in the algorithm at different search stages, which can significantly improve the algorithm performance. Ming *et al.* [61] proposed a simple and effective method called Pareto adaptive PBI (PaP) to adjust the penalty factor online according to the approximated PF shape and PBI decomposition method contours. Specifically, an appropriate  $\theta$  value is identified from a set of factor candidates, i.e., the optimal solution obtained by the PBI method with this value is closer to the weight vector compared with the same method with other  $\theta$  values. The effectiveness of the PaP method was verified on a real-world problem, i.e., the design of a hybrid renewable energy system with an unknown PF.

Another improvement over the original decomposition method is to reduce the volume of the improvement region, which is determined by the current solution, the optimal solution, and the contour of a subproblem, thus promoting population diversity. In Refs. [17] and [62], the improvement region of the solution  $x^i$  for the subproblem  $i$  in the objective space was defined as the set of all solution vectors that are better than the  $x^i$  vector, and indicated that a larger one can decrease population diversity, i.e., a better new solution could replace several different old solutions, resulting in the same solution for several different subproblems. To address this issue, various improved MOEA/D algorithms were proposed. Wang *et al.* [17] proposed an adaptively constrained decomposition (ACD) approach, which adds constraints on unconstrained subproblems and uses a parameter  $\theta$  to control the improvement region size of each subproblem, making it much smaller than the original one. Also, the approach adaptively adjusts  $\theta$  to maintain a balance between population diversity and convergence. Cai *et al.* [62] presented a constrained decomposition with grids (CDG), in which the objective space is divided into multiple grids. Each subproblem has a grid coordinate and a neighbor defined by the grid distance between two solutions, which is simple and intuitive. The grid system is renewed by updating the ideal point and nadir point to evolve the solution mapping vectors toward the PF. The proposed decomposition method can effectively reduce the improvement region and has strong robustness to the PF shape.

Combining the mathematical idea with the original decomposition method, Ma *et al.* [18] proposed Tchebycheff decomposition with  $L_p$  norm constraint ( $p$ -TCH) on the direction vector, where the objective function of each subproblem has clear geometric properties. On the basis of such properties, a generalized subproblem objective function can be defined, and it is then possible to adjust the weights or significances of subproblems in competition, mainly by changing the  $p$  value to accommodate the preference region of each subproblem. Therefore, more update chances can be assigned to some subproblems to which users pay more attention, accelerating their convergence.

Jiang *et al.* [19] presented two new decomposition methods with adjustable contours or improvement regions, namely the

multiplicative scalarizing function (MSF) and penalty-based scalarizing function (PSF), both of which use a parameter  $\alpha$  to control the improvement region of each subproblem. The influence of  $\alpha$  on the improvement region was investigated through simulation experiments, and the results showed that the size of the improvement region decreases with the increase of  $\alpha$  and both methods degenerate to TCH when  $\alpha = 0$ . Population diversity can be well guaranteed by changing  $\alpha$  for different subproblems.

## 2) COMBINATION OF THE ORIGINAL DECOMPOSITION METHODS

Each decomposition method used widely has its own advantages and disadvantages. Therefore, it is a simple and effective strategy to combine the complementary advantages of different decomposition methods, which makes the algorithm obtain more solutions.

TCH usually performs well on MOPs with convex and nonconvex PFs, while WS is only suitable for the optimization problems with convex PFs. Further, WS converges faster than TCH and gets a smoother PF. Ishibuchi *et al.* [20] used both WS and TCH for fitness assessment, which is mainly realized by two mechanisms based on a grid that consists of weight vectors. One mechanism assigns separate decomposition methods to different grids, i.e., all weight vectors within the same grid use the same decomposition method. The other mechanism alternately uses one of these two different decomposition methods to each of all weight vectors in a single grid.

Wei *et al.* [63] presented a weighted mixture-style (WM) decomposition method that combines the advantages of WS and TCH, which allows the algorithm to obtain more solutions. WM decomposition method was adopted in MOEA/D to transform an MaOP into a set of single-objective subproblems and to solve them in parallel. The experimental results showed that the WM decomposition method outperforms the three typical methods, i.e., WS, TCH, and PBI under the framework of MOEA/D.

WS and TCH are unsuitable for handling some MOPs with very different objective sizes. A direction-based decomposition method, termed normal boundary intersection (NBI) approach [21], is relatively insensitive to the scales of objective functions. Therefore, Zhang *et al.* [22] suggested a new decomposition method, called NBI-style Tchebycheff, which fully combines the characteristics of TCH and NBI. The algorithm that integrates this decomposition method into MOEA/D was presented, which finds two extreme points with minimum function values and then lets the solution mapping vectors approximate PF by minimizing the maximum distance between the points on the line composed of two extreme points and the point (solution) on the weight vector. Finally, the experimental study on portfolio optimization of real-world problems demonstrated the effectiveness of the proposed algorithm.

### 3) ADAPTATION OF THE ORIGINAL DECOMPOSITION METHODS

The introduction of adaptive strategies or learning mechanisms into MOEA/D makes it more suitable for solving complex problems. Adaptively selecting decomposition methods according to the problem characteristics is helpful to accelerating algorithm convergence and to obtaining more solutions dispersed widely.

Ishibuchi *et al.* [23] designed a mechanism for automatic selection between WS and TCH, which uses TCH only when the current solution mapping vector is close to the nonconvex regions of the PF, i.e., a current individual and at least  $T$  neighbors in its neighborhood have the same objective vector. This mechanism employs WS during the MOEA/D execution for the other solution mapping vectors around the convex regions. Since the PF is usually unknown before we find POSs, a method of detecting the nonconvex regions of the PF was proposed to determine whether TCH is required. TCH will be used when a single individual becomes the best with respect to the weighted sum under different weight vectors.

It is helpful for an algorithm to combine modules with different functions together. Wu *et al.* [24] proposed a learning-to-decomposition (LTD) paradigm including a learning module and an optimization module to adaptively set the decomposition method. Specifically, the learning module takes nondominated solutions from the optimization module as the training data, and then uses the Gaussian process regression to learn the characteristics of the estimated PF. According to the analytical model extracted from the learning module, the optimization module that adopts MOEA/D adaptively sets the decomposition method, including effective reference points and appropriate subproblem formulations. The validity of the LTD model was proved by a series of experiments on the benchmark problems with different PF shapes. This research provides an inspiration that the organic combination of MOEA/D and machine learning algorithms can make better use of the information generated in evolution, making the learning and optimization processes interact and benefit each other.

Wang *et al.* [64] systematically analyzed the searching ability of a family of scalarizing methods, the  $L_p$  scalarizing methods, where  $L_{p=1}$  represents WS and  $L_{p=\infty}$  corresponds to TCH. Simulation results showed that the probability of finding a better solution decreases with an increase of the  $p$  value, and the difference in the improvement region size for various  $L_p$  scalarizing methods becomes smaller when the solution mapping vectors approach PF. Therefore, the  $p$  value is crucial to the balance between convergence and diversity for a specific problem. A Pareto adaptive scalarizing strategy was then proposed to assign an appropriate  $L_p$  scalarizing method for each weight vector during the search. To be more specific, an appropriate  $p$  value is found from the set of different  $p$  values when using the  $L_p$  scalarizing method, i.e., the optimal solution obtained with this  $p$  value is closer to the search direction compared with other  $p$  values.

### 4) METHODS BASED ON OBJECTIVE REGION DECOMPOSITION

Each method of this classification does not require any aggregation functions and only needs the DM to select a group of direction vectors, resulting in less human labor compared to the decomposition method used in the original MOEA/D framework. Liu *et al.* [25] proposed a new version of MOEA/D, MOEA/D-M2M. Unlike MOEA/D decomposing an MOP into several single-objective subproblems, this algorithm decomposes the target MOP into a set of simpler multiobjective optimization subproblems, known as multiple to multiple (M2M). The objectives of each subproblem and original problem are just the same, while their feasible regions are different. Each subproblem has its own population to evolve its PS, and the PSs of all subproblems constitute the PS of the original problem. The PF of each subproblem has linear geometric shapes, which is applicable to the domination-based MOEAs for processing MOPs with simple PS shapes.

The MOEA/D-M2M framework decomposes the objective space and cooperatively evolves all the subpopulations, each of which corresponds to a specific subproblem, effectively alleviating the difficulty of the traditional MOEA/D in solving MaOPs with the high-dimensional objective space. Therefore, more information on MOEA/D-M2M is given in the fourth subsection (III.D), many-objective optimization.

### 5) SUMMARY

The decomposition methods introduced in this subsection are summarized in Table 2, from which we can obtain the following key findings.

- Many studies [23], [60], [61] introduce adaptive mechanisms into MOEA/D, which adaptively adjusts the parameters of the methods proposed in previous studies, and balances population diversity and convergence of each algorithm at different search stages.
- There is a highly promising prospect if we make full use of the advantages of different decomposition methods by using mixed mechanisms [20], [22], [63].
- It is helpful to enhance the solution quality by improving original decomposition methods in terms of properties of decomposition methods such as ideal points [16] and contours [17], or proposing new mechanisms with mathematical methods [18].

### B. WEIGHT VECTOR GENERATION METHODS

The weight vector defines each subproblem that maintains the optimal solution during an evolution, and its generation method affects not only the distribution of Pareto optimal vectors in the objective space but also the execution speed of an algorithm. There are two main drawbacks of the simplex lattice method in the original MOEA/D: a) zero weight vectors affect the PS quality, and b) the distribution of weight vectors is uneven because there are too many boundary weight vectors in the experimental field. Thus, quite a few methods



TABLE 2. Summary of studies on decomposition methods.

Classification	Ref.	Decomposition method	Main idea(s)
Improvement or innovation	[16]	IPBI	Evolves solutions from the worst objective vector in the current population by maximizing the scalarizing function value.
	[17]	ACD	Adds constraints on subproblems to reduce the improvement region of the decomposition method and adaptively adjusts constraints.
	[18]	$p$ -TCH	Adds $L_p$ norm constraints to the direction vector, making the objective function of each subproblem have clear geometric properties.
	[19]	MSF and PSF	Presents two new decomposition methods, namely MSF and PSF, both of which use a parameter to control the improvement region of each problem.
	[60]	Adaptive PBI	Uses two new penalty schemes of APS and SPS to adaptively adjust the penalty factor of PBI at different search stages.
	[61]	PaP	Adaptively sets the penalty value based on the approximated PF shape and the PBI method contours.
	[62]	CDG	Divides the objective space into a plurality of grids, and makes the solution position approach to PF by updating the grid system.
Combination	[20]	WS + TCH	Uses WS and TCH for a fitness assessment.
	[22]	NBI-style TCH	Combines TCH approach and NBI approach that is relatively insensitive to the objective function scale.
	[63]	WM	Combines the advantages of WS and TCH and proposes a WM decomposition method to solve MaOPs.
Adaptation	[23]	WS or TCH	WS is used for individuals in the convex PF region, while TCH is only applied to individuals in the nonconvex PF region.
	[24]	LTD	Adaptively sets the decomposition method including reference points and subproblem formulations by learning the characteristics of the estimated PF.
	[64]	$L_p$ scalarizing	Analyzes a family of scalarizing methods, i.e., $L_p$ scalarizing methods, and proposes a Pareto adaptive scalarizing strategy to optimize the $p$ value.
Without any decomposition method	[25]	M2M	The target MOP is decomposed into $K$ multiobjective optimization subproblems by DM with $K$ unit vectors, and each subproblem has its own population.

for appropriately generating or adjusting weight vectors have been studied to overcome these defects, mainly including the following four categories.

### 1) METHODS BASED ON UNIFORM DESIGN

The goal of these methods is to make weight vectors be distributed more uniformly in the objective spaces. Zhang *et al.* [26] introduced a uniform design method into MOEA/D and named the resulting algorithm as MOEA/D-UD, which initializes the population and weight vectors based on the mixed uniform experiment, and could uniformly explore the region of interest (ROI) of DMs from the initial iteration. The proposed weight vector adjustment strategy makes full use of the information on neighboring individuals to identify crowding regions and sparse regions for the complex PF, and then a PF distributed uniformly is found by removing weight vectors from and adding weight vectors into two regions, respectively. Compared with three outstanding algorithms, namely, NSGA-II, MOEA/D-DE, and MOEA/D-AWA (adaptive weight vector adjustment) on 19 test instances, MOEA/D-UD can obtain a well-converged and diversified solution set within an acceptable execution time.

To solve the 0/1 knapsack problem that has been proved to be NP-complete, Tan *et al.* [65] embedded the so-called good lattice point (GLP) in the MOEA/D framework, and termed

the resulting algorithm as uniform design multiobjective evolutionary algorithm based on decomposition (UMOEAD), which determines a set of weight vectors distributed uniformly and thus disperses the decomposed scalar optimization subproblems evenly. GLP first constructs a lattice point set  $S_N$  with the smallest discrepancy among all possible generating vectors over the design ROI. Then  $N$  weight vectors scattered uniformly are obtained by using  $S_N$  under certain restrictions. Experimental results indicated that the proposed algorithm significantly outperforms the NSGA-II, SPEA2, and PESA.

### 2) METHODS BASED ON ADAPTIVE ADJUSTMENT

In real-world problems, the search complexity of each subproblem is usually different, and thus, weight vectors need to be dynamically adjusted according to the computational resources required by each subproblem.

Harada *et al.* [27] improved MOEA/D by adding a method of adaptively assigning weight vectors to subproblems. The method identifies the subproblems that are hard to search by using the EP and weight vectors, and then divides each of these subproblems into several subproblems to assign multiple weight vectors, implying that more computational resources are allocated in the direction of the subproblem with search difficulty. Therefore, the search speed of each

subproblem can be adjusted reasonably to find a wider and more uniform PS distribution. Li *et al.* [66] proposed an improved MOEA/D with a weight vector adaptation (WAD) strategy, which first identifies weight vectors that waste computational resources. Specifically, the resulting solution mapping vectors of their corresponding subproblems may fall into local optimum or point to discontinuous PF regions. Then the strategy adjusts these weight vectors and optimizes their computational resource allocations to accommodate specific MOPs.

For MOPs with concave PFs, Jiang *et al.* [28] proposed a novel method, termed Pareto-adaptive weight vectors ( $pa\lambda$ ), to automatically adjust the weight vectors according to the geometrical characteristics of the PF. This method uses the mixed uniform design to generate an arbitrary number of initial weight vectors with gradients denoted by  $\lambda$  lines, which produce  $N$  intersection points along the PF. Then weight vectors are adjusted so that the  $N$  intersection points have the maximum hypervolume to realize their uniform distribution, where the hypervolume is used to evaluate the distribution of these points and to drive the  $pa\lambda$  method.

When MOPs have an irregular PF, the fixed weight vectors cannot guide the algorithm to obtain the solutions distributed uniformly and widely. Particularly, in the discontinuous parts of a PF, several subproblems have the same optimal solution. While in the peak or low-tail region of a PF, many solution mapping vectors are distributed in a narrow region of the objective space. Qi *et al.* [29] proposed an improved MOEA/D algorithm with AWA. The AWA strategy identifies real sparse regions instead of discontinuous parts of the complex PF, and then introduces the elite population that helps add new subproblems in real sparse regions and remove redundant subproblems from the crowded parts of the PF. The sparseness of a subproblem in the AWA method is determined by  $m$  (the number of objectives) nearest neighbors of its solution, which results in the AWA being inaccurate when the solution of a subproblem has several close neighbors within the objective space in similar directions. To overcome this deficiency in [29], Qi *et al.* [67] defined a new neighborhood relation between subproblems by using the Delaunay triangulation net. Then the sparseness measurement of a subproblem is determined by not only Euclidean distances between solutions but also their distribution.

### 3) METHODS BASED ON USER PREFERENCES

The algorithm of this category uses the mechanism that is dependent on user preferences to objectives. Pilát and Neruda [30] presented an MOEA/D with the co-evolution of weights (cwMOEA/D), which introduces the CE method to adaptively adjust weights in terms of user preferences during an evolution. DMs specify their preferences by assigning the binary preference values to the individuals in a form of function. After every iteration, the weight vectors are adjusted adaptively by the CE method. The update process generates new weights by the Gaussian mutation with different standard deviations, refines the user preferences function,

assigns individuals with both the parent weights and the offspring weights, compares the preferences of the assigned individuals, and selects the weight that corresponds to the individual with a lower preference. The experiments showed that the cwMOEA/D is able to find well-spread solutions in the preference regions specified by DMs, thus effectively utilizing the computational resources.

Ma *et al.* [13] proposed an MOEA/D with the biased weight vector (pMOEA/D), which adjusts the distribution of weight vectors based on the geometric analysis of a modified TCH decomposition method for handling the MOPs with preference information. This pMOEA/D attempts to maintain a set of nondominated solutions in the DM's ROI, rather than all solutions of a PS. Particularly, some subproblems whose solutions remain far away from the preference regions are removed while some new subproblems that may help search the preference regions are added into the current evolutionary population. The capability of pMOEA/D to solve practical engineering problems was investigated by testing two multi-objective reservoir flood control problems.

### 4) OTHER WEIGHT VECTOR GENERATION METHODS

There are some weight vector generation methods that are hard to categorize. The fixed weight vectors in MOEA/D may not work well when the PF of an MOP is irregular or complex. An MOEA/D with both fixed and random weight vectors (MOEA/D-RW) was suggested in [31]. The random search direction (weight vector) is considered to optimize the subproblems only if the solutions to all subproblems with fixed weight vectors have no improvement over several iterations. The proposed algorithm was compared with MOEA/D-DE and NSGA-II on a number of benchmark problems with irregular PFs, and results showed that it outperforms the two other algorithms.

Applying the idea of hierarchy structure of a social organization to MOEA/D, Xu *et al.* [68] proposed a new MOEA based on hierarchical decomposition (MOEA/HD). The algorithm builds a superior-subordinate relationship for all subproblems, and then layers subproblems into two different hierarchies, higher- and lower-level ones. In addition, according to search results of higher-level subproblems, the search direction of lower-level subproblems is adaptively adjusted in terms of the vertical bisector between the current solutions of two higher-hierarchy subproblems.

### 5) SUMMARY

Table 3 outlines the main points of weight vector generation methods discussed in this subsection. From the different works addressed above, we remark that the weight vector guides the search direction of each algorithm, and the solutions distribution depends highly on the weight vector generation methods. Therefore, the appropriate methods should be adopted to generate uniform or uneven weight vectors that suit specific problem features or user requirements well.

If the PF shape is regular or known in advance, weight vectors generated by the uniform design method can make

**TABLE 3. Summary of studies on weight vector generation methods.**

Classification	Ref.	Algorithm	Main idea(s)
Uniform design	[26]	MOEA/D-UD	Introduces a uniform design method into MOEA/D to initialize the population and generate weight vectors.
	[65]	UMOEAD	Uses GLP uniform method to find a set of weight vectors distributed uniformly with the lowest discrepancy in the design ROI.
Adaptive adjustment	[27]	Adaptive weight vector assignment	Divides the subproblem that is difficult to search into several subproblems to assign multiple weight vectors, thus allocating more computational resources in the direction of this difficult subproblem.
	[28]	$p\lambda$ -MOEA/D	Automatically adjusts the distribution of weight vectors according to the PF geometrical shapes.
	[29]	MOEA/D-AWA	Removes some subproblems from the crowded regions of the PF and adds other new subproblems in its real sparse regions.
	[66]	MOEA/D-WAD	Identifies the weight vectors that waste computational resources and adjusts them to optimize computational resource allocations.
	[67]		Improves [29] and defines a new neighborhood relation by using the Delaunay triangulation net to determine the sparseness of a subproblem.
User preferences	[13]	pMOEA/D	Integrates biased subproblem adjustment into MOEA/D to search the DM's ROI.
	[30]	cwMOEA/D	Adjusts weight vectors adaptively by using the CE method that incorporates binary user preferences.
Others	[31]	MOEA/D-RW	Proposes a modified version of MOEA/D with both fixed and random weight vectors.
	[68]	MOEA/HD	Layers subproblems into different hierarchies, and adaptively adjusts the search direction of lower-hierarchy subproblems based on higher-hierarchy search results.

the distribution of solutions more uniform. The adaptive adjustment strategy is more suitable for MOPs with complex PFs, which may have discrete regions, sharp peaks, and a long low-tail. According to the geometrical characteristics of the PF or the search difficulty of each subproblem, the weight vector is adjusted adaptively to realize the reasonable allocation of computational resources and to make solution mapping vectors be distributed more evenly in the objective space. Also, user preferences can be used as a guidance to adaptively assign the weight vector to each subproblem, aiming at providing diverse preference solutions.

### C. EVOLUTIONARY OPERATORS

The evolutionary operators of mating selection, reproduction operators, and replacement mechanisms are the main drivers for EAs to generate promising offspring, which are discussed in detail below.

#### 1) MATING SELECTION

The original MOEA/D has a neighborhood-based mating pool, which is not conducive to generating more new solutions in the early search. Facing this deficiency, Li and Zhang [32] allowed each individual to mate with any others in the entire population with a probability  $(1-\delta)$  to promote population diversity. Chiang and Lai [33] adopted the mating selection strategy proposed in [32], but it only works for the individuals with unresolved subproblems and allocates more computational resources in their search direction. It is generally considered that a subproblem is resolved if its solution has not been improved for  $\alpha$  consecutive generations. However, the solution improvement of a subproblem may stop temporarily while it continues later. Therefore, it is not enough to judge the unresolved subproblems only

once. Besides, an adaptive mating selection mechanism was designed to dynamically adjust the mating pool of each individual according to the Euclidean distance between individuals in the decision space instead of the distance between weight vectors of their subproblems in the objective space.

In addition to changing the range of mating selection, the selection technique has been improved. Many scalar subproblems may obtain similar solution mapping vectors on a PF, leading to a decline in population diversity. To deal with this issue, Jiang and Yang [34] introduced a new niche-guided scheme into the MOEA/D framework to perform the mating selection in the less crowded regions of the PF. Particularly, the scheme computes a neighbor-related niche count of each individual. If the count becomes greater than a threshold given, which means that the individual is similar to its neighbors, the parents for mating should be selected from outside the neighborhood. Finally, the population diversity is successfully improved by reducing the probabilities of duplicated solutions in the offspring.

#### 2) REPLACEMENT MECHANISM

In the original MOEA/D, a superior offspring solution could replace several inferior neighboring solutions, leading to the deterioration in population diversity. Wang *et al.* [17] proposed a constrained decomposition approach that imposes some constraints on unconstrained subproblems to reduce the replacement range, and further developed a new strategy to adaptively adjust constraint factors by using information collected from the search, which is helpful to balancing population diversity and convergence at different search stages for an algorithm.

It is uncertain whether the new solution  $x_{new}^i$  of the subproblem  $i$  is most suitable for its neighboring subproblems  $B(i)$ , i.e.,  $x_{new}^i$  may be bad for  $B(i)$ , while good for other subproblems. It is thus unreasonable to directly replace the solutions of  $B(i)$  with  $x_{new}^i$ . Wang *et al.* [35] investigated a global replacement (GR) strategy. It finds the most suitable subproblem  $j$  for the new solution  $x_{new}^i$  by comparing the objective function value of each subproblem, and then selects  $T$  closest subproblems to subproblem  $j$  to constitute the replacement neighborhood  $B(j)$ , which decreases the probability of replacing multiple current individuals with one generated newly. However, the replacement neighborhood is set to a fixed size in the experiment, which is not suitable for complex MOPs. Ref. [36] extended the GR strategy and presented an approach to dynamically adjust the replacement neighborhood size, which can effectively control the tradeoff between convergence and diversity at different search stages of an algorithm.

The mutual selection between promising solutions and subproblems during the replacement process can be regarded as a match between solutions and subproblems. However, the solution does not explicitly express a preference for subproblems in MOEA/D. Li *et al.* [69] suggested the use of a simple and effective stable matching model to coordinate the selection process. Each subproblem ranks all the solutions including its parent and offspring according to its aggregation function and prefers the solutions with better function values. Besides, each solution ranks all subproblems by calculating its distance to the direction vectors of these subproblems and prefers the subproblems whose direction vectors are close to it. Finally, the model matches each subproblem with a single solution.

### 3) MATING SELECTION AND REPLACEMENT MECHANISM

Different from just improving either mating selection or replacement, it makes sense to conduct both in one algorithm to further raise algorithm performance. The neighborhood structure should be independent when choosing mating parents and replacing old solutions. Also, different MOPs or a specific problem may need different neighborhood sizes, and thus, adaptive neighborhood sizes firmly contribute to the performance improvement of an algorithm. Ishibuchi *et al.* [70] introduced two neighborhood structures, i.e., the mating neighborhood for parent selection and the replacement neighborhood for interaction among individuals. The effect of neighborhood size on the algorithm performance was also studied, and a better result could be obtained by combining a small permutation neighborhood with a large matching neighborhood. The mating neighborhood is the entire population, while the replacement neighborhood size should be set to be much smaller than that of the original neighborhood in [32].

To overcome the shortcomings of the fixed neighborhood size in MOEA/D, Zhao *et al.* [71] suggested an ensemble of different neighborhood sizes in MOEA/D and adaptively adjusted their selection probabilities based on

their historical information. During an evolution, each subproblem selects one candidate from the pool containing  $K$  fixed-size neighborhoods based on the candidates' previous performance of generating improved solutions. Experimental results demonstrated that the mechanism combining adaptation and optimization is effective for improving the performance of multiobjective optimization algorithms.

### 4) REPRODUCTION OPERATORS

Reproduction is a direct and effective way to generate new solutions. Li and Zhang [32] adopted DE as a crossover operator and proposed an MOEA/D-DE to deal with continuous MOPs with complex PSSs. According to the characteristics of the DE operation, three parent solutions should be selected from the mating pool, which can produce a wide range of child solutions to effectively maintain population diversity. Zamuda *et al.* [37] extended the work on DE in [32] and introduced adaptive DE and local search into MOEA/D for solving constrained multiobjective optimization problems (CMOPs). Each individual has two parameters, self-adaptive  $F$  (amplification factor of the difference vector) and  $CR$  (crossover control parameter). The algorithm adjusts both to the appropriate values during an evolution, resulting in faster convergence than the original DE. Local search only performs on partial individuals that have improved since the last local search. The combination of global search and local search could enable the improved algorithm to maintain a good balance between exploration and exploitation.

As one of swarm intelligence algorithms, ant colony optimization (ACO) can achieve segmentation estimation of a PF by exploiting cooperative evolution among different ant groups. To avoid heavy computational overhead and accelerate algorithm convergence, Ke *et al.* [38] proposed an MOEA/D-ACO algorithm in which an ant is considered as a basic evolution unit. The  $N$  ants are divided into  $K$  groups by clustering their corresponding weight vectors, and each group is designed to approximate the small range of a PF. Different ant groups co-evolve with the POSs via sharing a global solution pool. Each ant records the best solution found so far for its subproblem and constructs a new solution with the pheromone matrix about itself and its neighbors. Then the algorithm performs solution updates like the original MOEA/D. This work has implied that many intelligent algorithms, such as particle swarm optimization, simulated annealing and artificial immune, can be combined with MOEA/D to further improve the algorithm performance.

Li *et al.* [72] proposed a bandit-based adaptive operator selection method to determine which reproduction operator should be selected in terms of operators' recent performances. To be more specific, this method uses the fitness improvement rate that refers to the solution quality difference between the parent and its offspring to calculate the reward of each operator, and sums all these rewards caused by its generated offspring solution replacing several parent solutions up as the final reward. The best operator is therefore selected from a group of operators for use at the next point time based on these

reward values. The idea of using cumulative improvement rates to evaluate the quality of operators in this literature is useful for some online selection problems, such as the allocation of different neighborhood sizes or population sizes to different search stages of an algorithm.

Ma *et al.* [73] incorporated an improved Baldwinian learning operator into the MOEA/D framework (MOEA/D-BL). The algorithm divides the current parent population into  $K$  disjoint clusters, and then obtains the evolution information by building a local distribution model for each cluster. Next, the Baldwinian learning operator constructs an offspring descent direction by combining the evolution history of the parent individuals with the distribution model. The combination of clustering and learning can make full use of the distribution information in the local region, which enables MOEA/D-BL to converge faster than MOEA/D.

Different operators have different search characteristics, and it is promising to make full use of these operators to balance convergence and diversity. Wang *et al.* [74] first explained the search behaviors of two different operators, namely, differential evolution operator and polynomial mutation, as well as neighbor learning operator and inversion mutation. Specifically, the former is more likely to find a set of solutions distributed widely and uniformly, while the latter has a good searching ability for higher quality solutions. Then the two operators were employed jointly in the MOEA/D-GR framework proposed by the authors to improve algorithm performance. Similarly, Li *et al.* [75] embedded both DE and covariance matrix adaptation evolution strategy (CMA-ES) in MOEA/D. To overcome the high computational costs of CMA-ES and to use the simplicity of DE, all subproblems are clustered into several groups, and each group selects only one subproblem to be solved with CMA-ES, while the others are solved with DE.

## 5) SUMMARY

As the main step in the MOEA/D framework and key procedure for each subproblem to produce the promising offspring, evolution is reviewed in this subsection, which includes mating selection, reproduction, and replacement. Table 4 presents the main idea, evolutionary operator and classification for each reference. Some valuable conclusions are given below.

- To maintain population diversity, it is promising to reasonably set the ranges of both mating selection and replacement, or to achieve a match between solutions and subproblems, which effectively reduces the probability that a new solution replaces old ones.
- According to the solutions distribution in an evolutionary process, the adaptive selection of different evolutionary operators or their combination with other heuristic algorithms, and even with hyper-heuristic algorithms, is beneficial to further improving the algorithm performance.
- Different evolutionary operators can be applied to different algorithm stages or populations to better

accommodate the different performance requirements of an algorithm.

## D. MANY-OBJECTIVE OPTIMIZATION

Unlike MOEAs based on Pareto dominance, the MOEA/D framework is not easily affected by selection pressure issues. Although MOEA/D is very effective in solving optimization problems with no more than three objectives, there are many challenges in applying it directly to solve MaOPs: 1) the number of weight vectors cannot be set arbitrarily, and it remains challenging to generate and initialize weight vectors at a low cost; 2) the population size grows nonlinearly as the number of objectives increases, resulting in high computational complexity; 3) failing to achieve a good coverage of a PF, i.e., maintaining population diversity presents certain challenges, mostly due to the properties of decomposition method contours; 4) offsprings are far apart from each other or from their parents in the high-dimensional objective space, which greatly weakens the effectiveness of evolutionary operators in producing promising offsprings; 5) the number of Pareto optimal vectors required to approximate the entire PF grows exponentially with the increasing number of objectives; and 6) the visualization of the high-dimensional PF is hard, which makes it more difficult for DMs to select a preferred solution. In order to overcome these challenges and improve the performance of the traditional MOEA/D in solving MaOPs, a large number of studies have been conducted, mainly involving the following four aspects.

### 1) STUDIES BASED ON DECOMPOSITION

For MaOPs, how to select a set of weight vectors to produce the expected distribution of PS is of great significance and challenge. Giagkiozis *et al.* [39], [76] defined a mathematical programming model with respect to the optimization of weight vectors. Different from a nonlinear one used for the TCH decomposition, the objective function in this model is simply a linear transformation of the weight vector, which means that the transformed vector is also a part of the convex set because of the convexity of the weight vector, and thus the model remains convex. On this basis, a generalized decomposition method was proposed, which improves the TCH approach and selects weight vectors by using the defined programming model. It produces the Pareto optimal vectors along the PF according to user preferences, and provides a framework in which DM can guide the basic EAs to a specific region or the entire region of the PF. Therefore, the generalized decomposition method can be more easily applied to a wide range of EAs. The significant advantage of this method is that it simplifies MaOPs by unifying three performance criteria (convergence, diversity, and coverage [2]) of multi-objective optimization algorithms into only convergence.

Facing the third challenge just mentioned, Yuan *et al.* [40] presented an MOEA/D variant with a distance-based updating (MOEA/D-DU) strategy, which fundamentally aims at clearly maintaining the solutions diversity during an evolution. Specifically, the  $K$  weight vectors closest to the new

**TABLE 4.** Summary of studies on evolutionary operators.

Evolutionary operators	Classification	Ref.	Main idea(s)
Mating selection (MS)	MS range	[32]	Each individual mates with any others in the entire population with a probability $(1-\delta)$ .
		[33]	Adopts ideas in [32] but the MS is performed only on unsolved individuals.
	MS technique	[34]	Uses a niche-guided scheme for handling MOPs with disconnected PFs, which selects parents in the less crowded region.
Replacement Mechanism (RM)	Replacement strategy	[35]	Proposes a GR strategy, which finds the most suitable subproblems for the new solution, thus reducing the replacement range.
		[36]	Extends [35] and dynamically adjusts the replacement neighborhood size.
	Replacement range	[17]	Adds constraints to subproblems to reduce the probability of replacing multiple old solutions with a new one.
		[69]	Uses a stable matching model to match each subproblem with a single solution.
MS and RM	Neighborhood	[70]	Defines independent neighborhood structures for MS and RM, but the neighborhood size is fixed in the experiment.
		[71]	Suggests an ensemble of different neighborhood sizes in MOEA/D and adaptively adjusts their selection probabilities based on their historical information.
Reproduction operators	MOEA/D-DE	[32]	Uses DE as the crossover operator, which selects three parent solutions and could generate a wide range of offspring solutions.
		[37]	Introduces adaptive DE and local search into MOEA/D for solving CMOPs.
	MOEA/D-ACO	[38]	Embeds an ant colony optimization algorithm in MOEA/D to avoid heavy computational overhead by co-evolving different ant groups, each of which estimates a small range of the PF.
	Self-adaption	[72]	Selects one reproduction operator online from a group of operators based on their recent performances for generating promising solutions.
	MOEA/D-BL	[73]	Embeds Baldwinian learning in MOEA/D, which constructs an offspring descent direction by combining the parent individuals' evolving history with the distribution model of a current population.
	Combination	[74]	Combines two different reproduction operators in the MOEA/D-GR framework to balance convergence and diversity.
[75]		Uses either DE or CMA-ES in the MOEA/D to optimize subproblems.	

solution  $S_{new}$  mapping vector are determined according to the vertical distance between  $S_{new}$  mapping vector and weight vectors in the objective space, and then their corresponding solutions are replaced with  $S_{new}$  by comparing aggregation function values. This differs from [77] that directly uses the vertical distance as a measure of the solution comparison. The solution update process of the proposed algorithm is stable, i.e.,  $S_{new}$  may replace only one old solution in the current population, resulting in a better PF coverage. In addition, the value of parameter  $K$  was determined experimentally. A larger  $K$  value puts more emphasis on the aggregation function for encouraging convergence, while a smaller  $K$  value highlights the vertical distance for promoting population diversity.

MOEA/D-M2M, first proposed by Liu *et al.* [25], is also suitable for solving MaOPs. In fact, many practical problems have  $m$  objectives, while the true PF is less than  $m$ -dimension due to the redundancy of some objectives. For the degenerated MaOPs with redundant objectives and very low-dimensional PS, the uniform and fixed weight vectors generated in [25] waste computational resources. Many researchers have improved the original MOEA/D-M2M with different mechanisms. Liu *et al.* [78] introduced an adaptive mechanism into the MOEA/D-M2M framework, which adaptively adjusts weight vectors of each subregion with the

Max-Min method according to the distribution information of the solutions found so far, thus enabling a region to be decomposed adaptively. Chen *et al.* [79] designed a new dominance relation called D-dominance, which combines the advantages of decomposition and domination. Then the D-dominance was employed into the MOEA/D-M2M as the selection criterion for each subpopulation. Besides, the resulting algorithm further utilizes the weight vector and reduces its design difficulty. Lin *et al.* [80] proposed an algorithm combining MOEA/D-M2M and  $I_{SDE+}$  (the indicator based on the shift-based density estimation) that contains both information on individual distribution and convergence. The algorithm calculates the indicator  $I_{SDE+}$  independently in each subregion, which can effectively maintain population diversity and reduce the computational cost.

Li *et al.* [81] combined the decomposition with the Pareto dominance for distinguishing and selecting candidate solutions, and termed the resulting algorithm as MOEA/D with domination archive (MOEA/DD). Each weight vector in this algorithm specifies not only a subproblem for fitness evaluation but also a subregion for estimating the local density of a population. In the original MOEA/D, the nondominated solutions are selected by comparing the aggregation function values between solutions, while in MOEA/DD they are selected according to both Pareto domination and crowding

distance sorting. In addition, the update mechanism of the parent population is stable, i.e., it considers only one offspring to update the population each time, and performs multiple rounds of update procedure if more than one offspring solution has been generated. To make parents closer to each other and thus improve the effectiveness of crossover operators, a subregion-based mating restriction scheme was introduced into MOEA/DD. By applying D-dominance to the MOEA/D-M2M framework, Chen *et al.* [82] proposed an adaptive D-dominance relationship-based evolutionary algorithm called DrEA. It employs a customized D-dominance with adjustable parameters to each subpopulation independently, which enables parallelism of D-dominance in each subpopulation. Therefore, the balance between population convergence and diversity can be achieved better compared with MOEA/D.

Ishibuchi *et al.* [83] demonstrated that the shape and the size of the PF have a large effect on the performance of the weight vector-based algorithms that perform well only when the distribution of weight vectors is consistent with the PF shape. When the PF shape is irregular, the design of weight vectors to ensure the diversity of solutions is challenging. He *et al.* [84] proposed a dynamical decomposition strategy for many-objective optimization. Different from existing decomposition methods, this method decomposes the objective space into subregions dynamically without employing a set of predefined reference vectors. Instead, the solutions themselves are considered as reference vectors. Thus, the performance of the proposed algorithm is less dependent on the PF shapes and remains robust especially in solving MaOPs with irregular PFs.

## 2) STUDIES BASED ON UNIFORM DESIGN

The challenges resulting from the weight vector generation and population size design in the high-dimensional space can be effectively overcome by the uniform design method. The motivation of uniform design here is to extend MOEA/D to many-objective optimization.

The method of simplex-lattice design used in the original MOEA/D cannot generate weight vectors with an arbitrary number, which restricts the extension of MOEA/D to solve MaOPs. To deal with this drawback, Tan *et al.* [41] proposed a transformation method and termed the resulting algorithm as UMOEA/D. A parameter  $M$  was designed to measure the nonuniformity, with a smaller  $M$  meaning that the weight vector distribution is more uniform. GLP was adopted to determine the weight vector set with a minimum  $M$ . The algorithm not only finds a set of weight vectors that are distributed evenly over the objective space, but also prevents the population size from becoming very large as the number of objective functions increases. However, when the size of the evolving population is small, UMOEA/D faces the risk of missing some boundary parts of the PF, especially for MaOPs, which can be effectively overcome by the hybrid method proposed in [42].

Simplex-lattice design in [3] assigns too many weight vectors at the boundary of the weight space. On the contrary, the transformation method [41] allocates no weight vector on the boundary of weight space and can generate any number of weight vectors. Ma *et al.* [42] proposed an improved MOEA/D with uniform decomposition measurement (UDM), which is called MOEA/D-UDM and combines two methods above to construct uniform weight vectors. Particularly, it first generates alternative weight vectors, most of which are located in the interior of the weight space with a few ones distributed on the boundary of the weight space. Then uniform weight vectors are selected based on the UDM mechanism from alternative weight vectors. Compared with the UMOEA/D [41], the proposed MOEA/D-UDM converges faster and the uniformity of obtained solutions is better.

Dai and Wang [43] proposed a new decomposition-based EA with uniform design (UDEA/D) to make the solution set have better convergence and diversity. The algorithm first generates uniform weight vectors with a uniform design method. Then, these weight vectors are used to divide the population into multiple subpopulations, each of which is associated with a subproblem, thus accelerating the algorithm convergence. In addition, a crossover operator based on uniform design is constructed to improve the searching ability of the algorithm. The authors also proposed the control of dominance area of solutions (CDAS) [44] to sort solutions of each subpopulation, guiding the search process to converge toward the PS. The simulation results indicated that the proposed algorithm has a good ability of exploration and exploitation as in [37], and outperforms the three famous algorithms, MOEA/D, NSGAII-CE and HypE (hyper evolving) on six benchmark functions with 5 to 25 objectives.

The design of weight vectors is a key issue for solving an MaOP with an incomplete PF, because the same POSs with different weight vectors were obtained in this case [83], thus greatly reducing population diversity. Gu and Cheung [45] developed a novel weight design method based on the self-organizing map (SOM), which periodically trains an SOM network with  $N$  neurons by using the objective vectors of the recent individuals, where  $N$  is the population size. The weights of neurons are employed as the weight vectors. This weight design method was then integrated into M2M and MOEA/D, respectively. Experiments on the MaOPs with an incomplete PF showed that these algorithms could generate weight vectors distributed evenly based on individuals' distribution, thus leading to a set of solutions distributed uniformly.

## 3) STUDIES BASED ON USER PREFERENCES

In the high-dimensional objective space, it is impossible to find all Pareto optimal vectors that approximate the real PF. Using preference information on reference points and reference vectors, the search range can be limited to the ROI rather than the whole objective space or more solution vectors can be searched in the ROI, which effectively overcomes the fifth-and-sixth challenges as stated in the beginning of subsection D. User preferences can be incorporated a priori,

posteriorly, or interactively into the search process. EAs belong to a posterior algorithm, i.e., DMs are not allowed to express their preferences until a PS has already been obtained, which is not conducive for DMs to making a comprehensive decision.

Based on their previous work, the R-MEAD (a reference point based multiobjective evolutionary algorithm with decomposition), Mohammadi *et al.* [46] developed a novel algorithm termed R-MEAD2. This algorithm utilizes weight vectors of decomposition methods to help handle user preferences by guiding solutions toward the preference region, potentially saving quite a huge amount of computational resources. Therefore, the algorithm is less susceptible to the selection pressure and converges to the PF more rapidly. For MOPs with highly nonlinear and complex PFs, a set of uniform weight vectors does not necessarily map to a set of uniform solution vectors in the objective space. Thus, the authors designed a feedback mechanism to adaptively update weight vectors based on the solutions distribution. Besides, a uniform random number generator was employed to decouple the population size from the number of objectives, making the population design easier.

Cheng *et al.* [47] defined the preference region by specifying a central vector and a radius, and designed a scalarization approach that is on the basis of the angle penalized distance and has a better capability in handling large-scale problems compared to the PBI approach. The resulting algorithm, called reference vector guided EA (RVEA), not only transforms the original MaOP into a set of single-objective subproblems, but also makes user preferences target the preference region of the PF. The distribution of reference vectors is also adjusted adaptively according to the range of objective function values. The experimental study demonstrated that the RVEA can make a solution set be distributed more uniformly when the PF has a regular geometric structure. However, it does not work well when the PF is irregular. Therefore, the authors proposed a reference vector regeneration strategy, which uses an additional reference vector set to explore the objective space further.

Li *et al.* [85] presented a nonuniform mapping scheme (NUMS), in which the original reference points distributed uniformly on the canonical simplex can be mapped to a new position close to the aspiration-level vector that is used to model the DM's preference information. Given the DM's requirements, the NUMS is able to obtain both a set of biased reference points toward the ROI and the ones located on the boundary, which enables MOEA/D not only to find preferred solutions but also to provide DMs with global information about the PF.

The above studies use reference points or reference vectors to represent user preferences. Li *et al.* [86] defined the target region that consists of the preferred range of each objective and used it to express preference information from DM. Then a preference-based algorithm of MOEA/D with a target region (T-MOEA/D) was proposed, which aims at well spreading nondominated solutions within the target region

and can be used a priori or interactively. Besides, to determine whether a solution is replaced or not, T-MOEA/D compares Pareto dominance and fitness values in turn, while the traditional MOEA/D only compares fitness values of solutions. Comprehensive experiments on a series of benchmark problems with 2 to 15 objectives fully demonstrated the effectiveness of the proposed algorithm, and more experimental problems should further be performed on different object regions.

Xiong *et al.* [87] constructed a preference model that combines the target region and reference point together, and introduced an interactive approach using the fuzzy theory to adjust the preference information during the search process for obtaining more reasonable solutions. This interactive method is beneficial to allocating more computational resources to the ROI of users. A tri-level ranking criterion was then designed to focus the search process on the preference region along with balancing convergence and diversity of a PS. Compared with the T-MOEA/D in [86], the proposed algorithm worked well in most of the test cases and showed superior performance in the convergence speed and computational time.

#### 4) OTHER RESEARCHES ON MANY-OBJECTIVE OPTIMIZATION

To address the third challenge, He and Yen [88] proposed a new algorithm, termed MOEA/D with diversity improvement (MOEA/D-DI). The algorithm consists mainly of two steps for different purposes. In the first step, a small number of Pareto solution vectors distributed uniformly on the PF boundary are generated, focusing on the algorithm convergence. The second step concerns population diversity. After finding the target solutions, a population is initialized around each solution, and the individuals in the population are well spread and distributed by using the proposed diversity method. In every iteration, two parents are selected from the whole population, and offsprings are generated after reproduction, and the next generation of individuals is then selected from the parent and its offspring. Similarly, Cai *et al.* [89] utilized two types of direction vectors, one aiming to approximate a more complete PF by expanding the number of direction vectors after the fast convergence along the boundary direction vectors, and the other trying to adjust the position of ineffective direction vectors for MaOPs with irregular PFs.

In the high-dimensional space, weight vectors in the weight space become sparse, resulting in insufficient search resources for each part of a PF. Sato *et al.* [90] introduced supplemental weight vectors and solutions into the original MOEA/D. The algorithm assigns supplemental weight vectors around each original weight vector and generates the corresponding supplemental solutions to enhance the detailed search in the objective space. Differing from the original MOEA/D, parents are selected from the supplemental solutions that have more similar objective values and variable information even in a higher dimensional objective space.



## 5) SUMMARY

In this subsection, we describe the main challenges that MOEA/D faces in solving MaOPs in the high-dimensional objective space. Then a series of improved algorithms to overcome these challenges are reviewed. Table 5 presents a summary of these algorithms based on their techniques, such as decomposition, uniform design, and user preferences-based.

The high-dimensional objective space makes it more difficult for MOEA/D to obtain a set of solutions distributed widely and uniformly. Objective region decomposition transforms an MaOP into several multiobjective subproblems and divides the population into multiple subpopulations, each of which solves a subproblem. Thus, the whole objective region can be well searched through CE among subpopulations, which greatly reduces the difficulty in searching over a high-dimensional objective space. Uniform design can effectively alleviate the difficulties of weight vector generation and population design. Using user preference information allows an algorithm to search only within the ROI of DMs and to generate a well-distributed solution set in the preference region, significantly saving the computational resources required for searching. Briefly, many-objective optimization presents one of the hotspots and challenges in future research and deserves further exploration.

## E. CONSTRAINED MULTIOBJECTIVE OPTIMIZATION

Many real-world optimization problems belong to CMOPs, such as constrained multiobjective portfolio optimization [48], resource-constrained unrelated parallel machine green scheduling problem [49], and risk-constrained energy and reserve procurement [50]. A CMOP usually includes multiple diversionary or conflicting objectives and different types of constraints that need to be satisfied simultaneously. These constraints lead to a number of infeasible solutions violating such constraints. Therefore, how to deal with the infeasible solutions generated during the search process and to realize the effective utilization of computational resources are the challenges when solving CMOPs. In recent years, many constraint handling techniques have been organically integrated into the MOEA/D framework to solve various CMOPs successfully. The following briefly introduces these techniques and their algorithms combined with MOEA/D.

### 1) PENALTY FUNCTIONS

The method based on the penalty function adds one penalty term to the objective functions, transforming the constrained optimization problem into an unconstrained one. In evolutionary computation, researchers choose the external penalty function method to solve the problem, mainly because it does not need to initially provide a feasible solution.

Jan and Zhang [51] presented a penalty function to punish infeasible solutions and introduced it into the update scheme of MOEA/D-DE. The proposed algorithm calculates the degree of constraint violation  $V(x)$  for the solution  $x$ , and

if  $V(x) = 0$ ,  $x$  is feasible, otherwise it is infeasible. The proposed penalty function uses the threshold related to the maximum and minimum values of  $V(x)$  to dynamically control the penalty amount which increases sharply when  $V(x)$  exceeds the threshold, encouraging the algorithm to search in both the feasible region and the infeasible region near the feasible one. The implementation process of the resulting algorithm is similar to that of MOEA/D, while has the following main variations: a) the calculation of  $V(x)$  for solution  $x$ ; b) using a new aggregation function (penalty function) to compare the new solution with the old one; and c) update of the  $V(x)$  value. The experiments showed that the algorithm performs well in six out of ten constrained multiobjective optimization test instances.

To keep a balance between objective minimization and constraint satisfaction, Fan *et al.* [52] presented a push and pull search (PPS) framework, and combined it with MOEA/D. As its name suggests, the proposed PPS divides the search process into two different stages of push and pull search. In the first stage, a CMOP is solved by considering only objectives without any constraints, which can help to get across to infeasible regions. In the pull stage, an adaptive penalty-based function is applied to pull the infeasible individuals achieved in the push stage to the feasible and non-dominated regions.

To improve the PPS framework [52] further, Fan *et al.* [53] proposed a self-adaptive penalty-based constraint handling method and embedded it in PPS. More specifically, inspired by the learning rate in deep learning, the penalty factor is dynamically adjusted according to the ratio of feasible solutions to a population, the constraint violation value, the target value, etc. In the original PPS, the Epsilon constraint handling method first considers constraint violation and then considers objectives. However, the proposed algorithm deals with objectives and constraints simultaneously, thus it can maintain better population diversity.

### 2) SEPARATION OF OBJECTIVES AND CONSTRAINTS

For the penalty function method, it is not easy to select the appropriate penalty factor for a specific problem, because the factor is too small to be an effective punishment and/or too large to enable an algorithm to search near the feasible region. The separation mechanism of objectives and constraints, as its name suggests, treats objectives and constraints separately, which does not require any additional parameters while they are needed for the penalty function.

Fan *et al.* [54], [55] improved two typical methods of this mechanism, epsilon constraint handling method and constrained dominance principle (CDP), respectively. More specifically, Ref. [54] dynamically adjusted the epsilon level, which is a parameter to control the relaxation of constraints in the epsilon constraint handling method. Then the improved approach was embedded in the MOEA/D framework, leading to a new algorithm called MOEA/D-IEpsilon. This algorithm compares the epsilon level of the newly generated solution with that of its neighborhood solutions and updates

TABLE 5. Summary of studies on algorithms for many-objective optimization.

Classification	Algorithm	Ref.	Main idea(s)
Decomposition	Generalized decomposition	[39] [76]	Produces the optimal vectors along the PF according to user preferences and provides a generalized framework.
	MOEA/D-DU	[40]	Proposes an MOEA/D variant with a distance-based updating strategy, which clearly maintains the solution distribution in evolution.
	Improved MOEA/D-M2M	[78]	Adaptively adjusts weight vectors of each subregion with the Max-Min method in the MOEA/D-M2M framework.
		[79]	Introduces a new dominance relation called D-dominance into MOEA/D-M2M as the selection criterion for each subpopulation.
		[80]	Combines MOEA/D-M2M and $I_{SDE}^+$ , and calculates the indicator $I_{SDE}^+$ independently in each subregion, which can effectively maintain population diversity and reduce the computational cost.
	MOEA/DD	[81]	Proposes a MOEA/D variant with domination archive, which combines the Pareto dominance with the decomposition.
		[82]	Applies D-dominance to the MOEA/D-M2M framework and employs a customized D-dominance with adjustable parameters to each subpopulation independently. Compared with MOEA/DD [81], the balance between population convergence and diversity can be achieved better.
Dynamical decomposition	[84]	Decomposes the objective space into subregions dynamically without employing a set of predefined reference vectors. Instead, the solution themselves are considered as reference vectors.	
Uniform design	UMOEAD	[41]	Proposes a transformation method to determine the uniform weight vector set with a nonuniformity measurement method.
	MOEA/D-UDM	[42]	Combines the simplex-lattice design with transformation method [41] to construct the weight vectors distributed uniformly via the uniform design measurement.
	UDEAD	[43]	Uses a uniform design method to generate weight vectors and constructs a crossover operator to improve the searching ability of the algorithm.
		[44]	Extends [43] and uses CDAS to sort solutions of all subproblems.
MOEA/D-SOM	[45]	Integrates a novel weight design method based on self-organizing map (SOM) into MOEA/D to generate weight vectors distributed evenly based on individuals distribution for uniform solution set.	
User preferences	R-MEAD2	[46]	Improves the previous work and presents a reference point based multiobjective evolutionary algorithm with decomposition (R-MEAD) for solving MaOPs.
	RVEA	[47]	Proposes a reference vector guided EA that makes user preferences target the preference region of the entire PF.
	NUMS	[85]	Uses a nonuniform mapping scheme to transform the original reference points distributed evenly into ones distributed prejudicially.
	T-MOEAD	[86]	Defines the target region as the preferred range of each objective and uses it to express user preferences.
	HMOEA-T	[87]	Presents a region preference-based many-objective evolutionary algorithm called HMOEA-T, which combines the target region and reference point together as the preference information adjusted by an interactive approach during the search process.
Other methods	MOEA/D-DI	[88]	Applies a diversity improvement method to MOEA/D, which initializes a population around each solution vector along the PF boundary.
	MaOEA/D-2ADV	[89]	Utilizes two types of direction vectors, one for approximating a more complete PF, and the other for adjusting the position of ineffective direction vectors for MaOPs with irregular PFs.
	Enhanced MOEA/D	[90]	Introduces supplemental weight vectors and solutions into the MOEA/D framework to enhance the solution search.

the population by replacing the inferior solutions. Then the epsilon level is updated according to the proposed equations. The comprehensive experimental results verified that the MOEA/D-IEpsilon outperforms the MOEA/D-CDP and the MOEA/D-Epsilon in the performance of both convergence and diversity.

Although the CDP is a popular constraint handling technique, it may neglect many infeasible solutions that have the potential to improve population diversity. Fan *et al.* [55]

proposed an angle-based constrained dominance principle (ACDP), which is specifically designed to solve CMOPs with a low ratio of feasible solutions to all ones. The ACDP adds the angle information of objective functions to the original CDP to enhance population diversity in the infeasible region. For the comparison between the feasible solution and the infeasible one, if the angle of the peer solutions remains less than a user-defined threshold, the feasible solution is better, otherwise they are nondominated. This differs from

the original CDP, which considers that any feasible solution is better than any infeasible solution. Experimental results indicated that the proposed ACDP method is very effective in the MOEA/D framework by comparing it with four popular constrained MOEA/Ds, including CMOEA/D, MOEA/D-CDP, MOEA/D-Epsilon and MOEA/D-SR (stochastic ranking).

Fan *et al.* [56] reviewed a number of popular constrained MOEAs based on decomposition, including MOEA/D-IEpsilon, MOEA/D-CDP, CMOEA/D that embeds the epsilon constraint handling approach in MOEA/D, and MOEA/D-SR that applies SR to handle constraints in MOEA/D. The comprehensive experimental results indicated that the MOEA/D-IEpsilon has the best performance in the decomposition-based CMOEAs.

### 3) REPAIR ALGORITHMS

An algorithm of this kind converts an infeasible solution to a feasible one by using repair operators. Fan *et al.* [57] proposed a new opposition-based repair operator to fix the infeasible solutions and then integrated it into two classical multiobjective EAs, MOEA/D and NSGA-II. This repair operator employs a reversed correction strategy that is inspired by the concept of opposition-based learning (OBL) [58]. The OBL method considers not only an estimate but also its corresponding opposite estimate, which can effectively avoid the search falling into local optimum. Finally, for the benchmark problems of CTPs and CMOPs, the algorithm with the proposed operator outperforms that with the two other kinds of repair operators used commonly in terms of convergence and diversity.

Repair methods are crucial to dealing with dynamic constrained problems since they help to move the infeasible solutions toward a feasible region. Ameca-Alducin *et al.* [59] investigated four repair methods with DE for dynamic constrained optimization, including reference-based repair, offspring-repair, mutant-repair, and gradient-based repair. Also, three different measures were proposed to compare the performance of these repair methods, and then their advantages and disadvantages were analyzed in terms of the experimental results. Although the literature studied the performance of various repair operators in the processing of the dynamic CMOP under the DE algorithm, these operators can still be applied to the MOEA/D, and the performances of resulting algorithms are compared as well.

### 4) SUMMARY

Each constraint handling technique has its pros and cons. The penalty function method is simple to implement, but the selection of penalty factors remains hard. The mechanism of separating objectives and constraints can be regarded as an extension of the traditional penalty function method, but it does not need to configure any parameters. Repair algorithms are suitable for some instances which have a low repair cost and a small search bias, while they have to be customized for different problems.

Any kind of constraint handling mechanism should achieve a balanced search in feasible regions and infeasible regions, which is actually a challenge faced by MOEA/D when dealing with CMOPs. Concisely, for different CMOPs, we need to design the appropriate constraint handling technique and integrate it into MOEA/D to obtain the optimal feasible solutions.

## IV. FINDINGS OF SURVEY

A systematic summary of current research directions, related technologies, and corresponding references of MOEA/D variants is provided in Table 6, where it is very clear to see the state-of-the-art investigations on MOEA/D. To make researchers catch each category of techniques with corresponding references in a limited space, the findings of this article are also illustrated in Fig. 4 in seven aspects with the brief discussion that follows, which hopefully presents readers a clearer understanding of the techniques used commonly in MOEA/D variants.

### A. ADAPTIVE MECHANISM

The adaptive mechanism that is suitable for complex MOPs can realize the dynamic adjustment of MOEA/D components, such as the decomposition method [23], weight vector [29], neighborhood size [71], and reproduction operator [72]. With the increasing focus on artificial intelligence (AI), algorithm automation will be a hotspot in the optimization community.

### B. COMBINATION

The combination of different methods [20], [22] or diverse operators [74] can fully utilize their complementary advantages to maintain the balance between diversity and convergence. Other heuristic algorithms (e.g., ACO [38]) or mathematical optimization tools (e.g., Cplex [91]) can be combined with MOEA/D to conduct both global and local searches in the objective space, and to ensure the quality and diversity of the solution set.

### C. UNIFORM DESIGN

The distribution of POSs obtained depends highly on the weight vector generation method. Uniform design methods like the good grid point can be used to initialize population [26], construct crossover operators [43], and generate weight vectors [65], making the solution mapping vectors be distributed more evenly in the objective space.

### D. USER PREFERENCES

EAs usually take a long time to solve problems, which is considered as a defect of these algorithms. Embedding user preferences in MOEA/D a priori or interactively can avoid unnecessary space searches and thus effectively shortens the operating time of an algorithm, where user preferences could be represented by reference points [46], reference vectors [47] or target region [86].

TABLE 6. Summary of typical works on MOEA/D.

Research direction	Classification	References
Decomposition methods	Improvement	[16]-[19], [60]-[62]
	Combination	[20]-[22], [63]
	Adaptation	[23], [24], [64]
	Multiple to multiple	[25], [78]
Weight vector generation methods	Uniform design	[26], [65]
	Adaptive adjustment	[27]-[29], [66], [67]
	User preferences	[13], [30]
Evolutionary operators	Mating selection	[32]-[34], [70], [71]
	Replacement mechanism	[17], [35], [36], [69]-[71]
	Reproduction operators	[32], [37], [38], [72]-[75]
Many-objective optimization	Decomposition	[39], [40], [76]-[84]
	Uniform design	[41]-[45]
	User preferences	[46], [47], [85]-[87]
Constrained Multiobjective Optimization	Penalty functions	[51]-[53]
	Separate objectives and constraints	[54]-[56]
	Repair algorithms	[57]-[59]

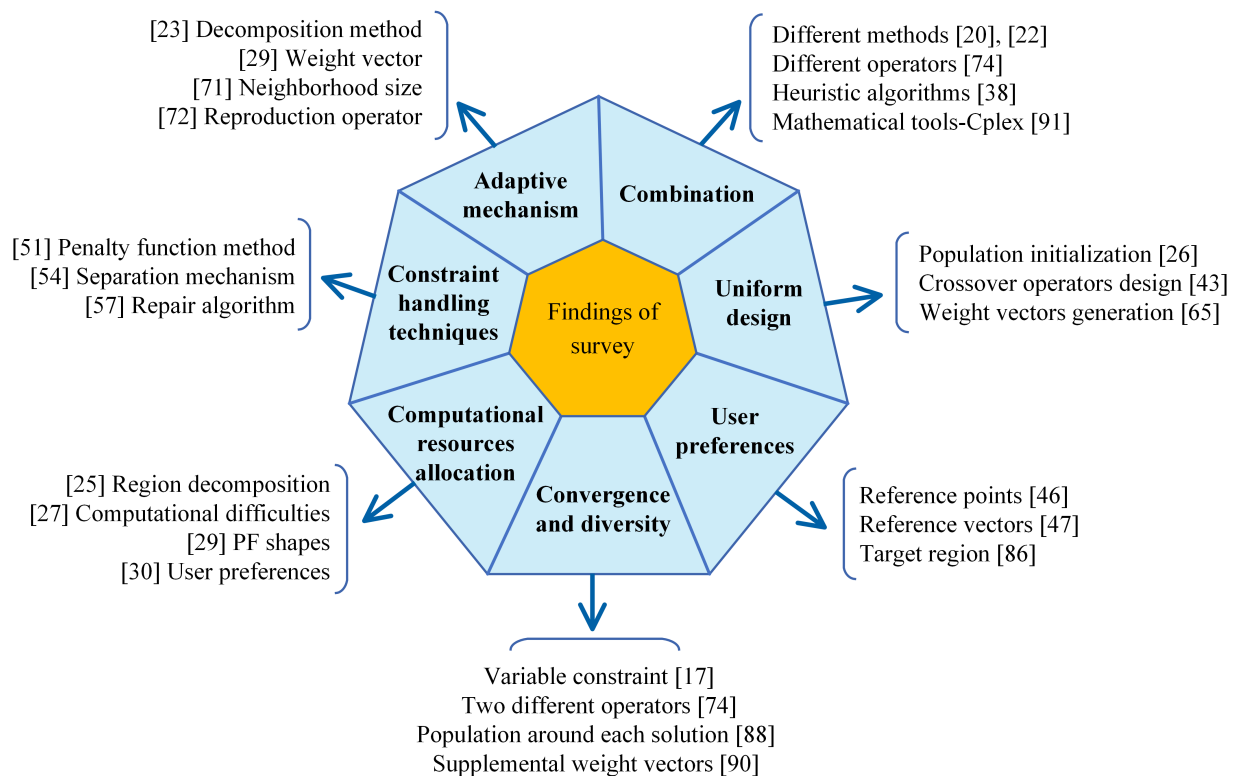


FIGURE 4. Findings of survey.

**E. CONVERGENCE AND DIVERSITY**

A lot of studies have been carried out on algorithm convergence and population diversity. Different constraint values are needed for different subproblems, even for a specific problem at different stages [17]. Two evolutionary operators with different search characteristics of the solution

uniformity and quality are used to generate promising solutions [74]. A population is initialized around each solution located on the boundary of the PS [88]. Supplemental weight vectors and the corresponding supplemental solutions are generated to enhance the search in a wider region [90].

## F. COMPUTATIONAL RESOURCES ALLOCATION

The computational cost increases sharply as the number of objectives increases and becomes extremely high in solving MaOPs. Therefore, it is significant to allocate computational resources reasonably. Many algorithms have been proposed to assign weight vectors that are used to allocate different amounts of computational resources for subproblems according to some criteria, such as computational difficulties of solving subproblems [27], geometrical characteristics of a PF [29], and user preferences [30]. To realize the adaptive decomposition for a region [25] and overcome the defect of wasting resources due to fixed and uniform weight vectors, another potential method is to assign subpopulations with different sizes to their corresponding subproblems.

## G. CONSTRAINT HANDLING TECHNIQUES

Various constraint handling techniques have been well integrated into MOEA/D to solve CMOPs. The penalty function method [51] is simple to operate, but it remains difficult to select the appropriate penalty factor for a specific problem. The mechanism that treats objectives and constraints separately [54] does not require any parameters, and thus it is more efficient and universal than the penalty function method. The repair algorithm [57] is another way to convert infeasible solutions into feasible ones, which can effectively avoid the search falling into the local optimum. However, each algorithm is always suitable for the specific problem only. Therefore, it is necessary to design suitable constraint handling techniques for different problems, which requires a comprehensive consideration of the characteristics of practical problems and algorithm implementation difficulty.

## V. CHALLENGES AND FUTURE RESEARCH DIRECTIONS

Despite the suitability, success, and prospect of MOEA/D for MOPs, many problems remain to be solved in both theoretical research and practical application. As shown in Fig. 5, we will discuss them and constructively propose some future research directions.

### A. MOEA/D THEORETICAL RESEARCH

#### 1) MOPS WITH AN UNKNOWN PF

When the PF is unknown beforehand, how to get a uniform and diverse solution set remains a challenging problem to which users often pay attention. It is promising to combine learning strategies (e.g., machine learning) with multiobjective optimization algorithms to learn and optimize interactively, so as to achieve complex PF prediction and adaptive adjustment. Particularly, machine learning is used to extract useful information from the data generated during an evolution, such as the solutions distribution and population clustering. Accordingly, these pieces of information and rules in the data can be fully utilized to guide MOEA/D for a more effective search.

#### 2) CONSTRAINED MULTI-OBJECTIVE OPTIMIZATION ALGORITHMS

One of the issues that may be encountered in solving CMOPs is how to deal with infeasible solutions. In addition to applying appropriate constraint handling techniques, designing special encoding and decoding methods may be promising to lower the possibility of generating infeasible solutions.

Another challenge in solving CMOPs is how to reasonably balance searches in feasible regions and infeasible ones. If the population diversity remains insufficient after entering the feasible region, the search will be concentrated on a certain part of the feasible region, thus making the solution fall into the local optimum. On the contrary, if the initial population is highly infeasible, then the algorithms may not converge to the feasible region. This challenge can be overcome by the following ways: a) defining an appropriate dominant mechanism and combining it with the constraint handling techniques as the criteria for selecting the nondominated solutions, and b) dividing one current population into many subpopulations, and using them to evenly search in both feasible region and infeasible region based on the current solutions distribution.

In the cases discussed above, the constraints remain unchanged in the optimization process. If the constraints change, how to deal with them presents an issue uncovered in many current studies, which is worth studying further.

#### 3) SINGLE-OBJECTIVE OPTIMIZERS

Meta-heuristic algorithms (such as tabu search, simulated annealing, and particle swarm optimization) or hyper-heuristic algorithms can be integrated into optimizers that are applied to solve single-objective subproblems. It is worth studying to dynamically update parameters of an optimizer according to information generated during an evolution, or to adaptively select various optimizers to better suit different problems. In addition, mathematical methods [92], such as operations research, convex optimization and dynamic programming, can be embedded in optimizers.

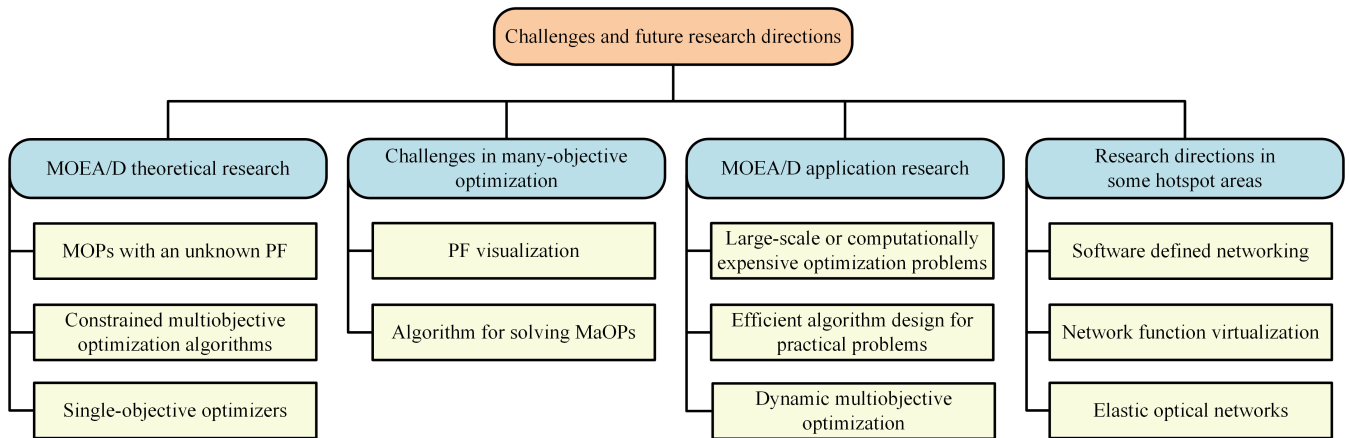
To maximize algorithm efficiency, various algorithms must be coordinated. Therefore, developing efficient parallel or distributed algorithms and proposing new hybrid algorithms are the two main research directions of MOEA/D in the future.

### B. CHALLENGES IN MANY-OBJECTIVE OPTIMIZATION

With the sharply increasing complexity of practical problems, many-objective optimization has been attracting researchers in the evolutionary multiobjective community in recent years. However, the relevant investigations are still relatively rare and many issues remain to be studied.

#### 1) PF VISUALIZATION

For the visualization of a PF in a high-dimensional space, most of the existing work shows that the indicator function value only varies along with the increase of the objective value, instead of displaying the location and distribution of



**FIGURE 5.** Challenges and future research directions.

the obtained solution vectors in a scatter plot, where each axis represents one objective. Facing this challenge, one of the future research directions is to apply some intelligent algorithms, such as big data and deep learning, to reduce the number of objectives while keeping the original objectives features unchanged. Then, one MaOP can be simplified to an MOP for considerable visualization, allowing DMs to easily select their preferred solutions. The idea of treating objective reduction as a multiobjective search problem in [93] and [94] is worth learning. For the MaOPs whose objectives could not be further reduced, the mapping transformation technique can be used to map solution vectors from the high-dimensional objective space to the two-dimensional or three-dimensional objective space, while preserving the dominance relationship among the solution vectors.

## 2) ALGORITHM FOR SOLVING MAOPS

A great deal of research work has been conducted on many-objective optimization, most of which aims at proposing mechanisms or methods for MaOPs with a large number of objectives, but little attention has been paid to MaOPs with many decision variables. Therefore, the algorithms for solving MaOPs in such a scenario need to be studied further. Ma *et al.* [95] proposed an MOEA based on decision variable analysis, which decomposes decision variables into several low-dimensional subcomponents, and thus a complicated MOP is decomposed into a set of simpler sub-MOPs. How to deal with mixed variables better is also an open problem. This literature inspires that we can use learning algorithms to analyze the characteristics of decision variables and the potential relationship between variables.

Due to the complexity of MaOPs, such as high-dimensionality, computational expense, and unknown function properties, designing or selecting an optimization algorithm for solving these hard problems is challenging. Ibrahim *et al.* [96] combined the benefits of several optimization algorithms and proposed a hybrid multi- and many-objective optimization algorithms framework to obtain the

fusion of solutions. Inspired by this idea, it is a promising future research direction to design a hybrid algorithm that adaptively selects a best-performing algorithm at different stages of the search process and executes all algorithms in parallel to reduce the computational cost.

In addition, many multiobjective optimization algorithms have only been tested on MOPs, and their performance should be further evaluated on various test problems with many objectives.

## C. MOEA/D APPLICATION RESEARCH

### 1) LARGE-SCALE OR COMPUTATIONALLY EXPENSIVE OPTIMIZATION PROBLEMS

MOEA/D has been widely used in the fields of engineering, industry, and science. However, it still faces challenges in solving large-scale or computationally expensive optimization problems. These problems involve both many objectives and large-scale decision variables, which leads to a rapid decline in EAs' performance when solving them. In recent years, some researchers have proposed various algorithms for overcoming this challenge.

Zhang *et al.* [97] proposed an evolutionary algorithm based on a decision variable clustering method that divides the decision variables into convergence- and diversity-related ones. These two types of decision variables were optimized with a convergence and diversity optimization strategy, respectively. It is still possible to improve the diversity of decision variable classification and the computational efficiency of the clustering method.

Wu *et al.* [98] presented a new hybrid algorithm including two phases of decomposition and optimization. The algorithm divides a large-scale problem into several small-scale ones. A modified self-adaptive discrete scan method and a hybrid search strategy were designed to solve the subproblems on the promising regions. Designing some effective decomposition methods for solving large-scale problems is valuable and challenging in the future.

Another challenge with such problems is that they require a large number of individual evaluations, resulting in a slow convergence speed of algorithms to solve them. Some research directions may be promising such as: a) using deep learning algorithms to acquire the features of large-scale problems, and then discarding redundant information to reconstruct the problem model [99], thus reducing its scale greatly; b) training the historical solution data to pre-select the better solutions to reduce the number of individuals for evaluation; and c) combining the high-efficiency parallelism of parallel computers with the natural parallelism of MOEA/D to accelerate the optimization process, such as the cooperatively distributed co-evolutionary process mentioned in [100], [101].

## 2) EFFICIENT ALGORITHM DESIGN FOR PRACTICAL PROBLEMS

Facing all kinds of practically complicated problems, how to effectively design algorithm components by making full use of the characteristics of a problem has a great influence on the quality of the final solution set. From the review above, it can be found that self-adaptation is a potential method for dynamically adjusting operators of all kinds, thus making the evolution toward the real PF. The divide-and-conquer algorithm transforms a big problem into several small subproblems, then tackles them one by one, and finally combines the solutions of all subproblems to solve the original big problem. Similar to this idea, the objective space can be decomposed into several small subspaces by multiple reference vectors, and then each of the multiple subpopulations is used to search within its corresponding subregion, and the solution of each subpopulation constitutes the final PS, which can effectively lower the search difficulty. In addition, we can integrate user preferences into MOEA/D a priori or interactively, reducing the search in unnecessary PF regions and accelerating the algorithm convergence.

## 3) DYNAMIC MULTIOBJECTIVE OPTIMIZATION

Due to scenario changes in objectives, constraints and parameters over time, dynamic multiobjective optimization problems (DMOPs) are increasingly a hotspot in the optimization field as such a real-world problem becomes more complicated, and how to track POSs that may not be the same as before efficiently is a key issue.

Some intelligent algorithms may be promising: a) algorithms designed by automation. These algorithms can flexibly add or remove some constraints and variables as the scenario changes, and be tested and improved through application experiments. At the same time, many modules in such an algorithm can realize the automatic selection of multiple candidate operators according to the evolution degree of the current solutions or scenario factors. b) prediction algorithms aiding population-based optimization methods. Because a new solution every time may be different from the previous optima, thus, prediction algorithms can be used to resume learning prediction models by training the historical POSs

under different scenarios to predict the optimal values in the future scenarios.

Concisely, combining MOEA/D with other intelligent technologies, such as learning and automation, to design more flexible and generalized algorithms is useful for solving DMOPs in the era empowered by AI.

## D. RESEARCH DIRECTIONS IN SOME HOTSPOT AREAS

Except for the three ones discussed above, there are some hotspot areas that have attracted both the academia and industry, such as SDN, NFV, elastic optical networks (EONs), edge computing [102], and 5G/6G networking. Due to limited space, we address the first three research issues for MOEA/D.

### 1) SDN

As an emerging network architecture, SDN has the distinct advantage of separating the control plane and data plane [103], resulting in flexible and efficient usage of network resources. It has found partial commercial applications and been tested extensively. However, it still faces many challenges in several aspects before its practical large-scale deployment, such as the software and hardware implementations in the SDN-based data plane, control plane, management plane and application plane. Some of the most important issues are the network planning for network elements designer, and the resource allocation algorithms for the operators in the scenario where there are hundreds of billions of terminals in the emerging Internet of Things.

In the case of constrained resources, the performance metric or multiobjective of the routing algorithm will be diversionary, such as the minimum objectives: capital expenditures (Capex), operating expenses (Opex), deployment cost, energy consumption [103], and end to end delay [104]. The maximum objectives are operators' profits, differentiated service levels, quality of experiences, etc. For example, S. Garg *et al.* [105] addressed an SDN framework for autonomous vehicles by using MOEA/D, aiming at minimizing the network latency and maximizing bandwidth utilization.

The architecture of numerous SDN controllers is reasonably a multi-layered tree since it matches the structure of our community well. We could use MOEA/D variants [78] to partition such huge terminals into thousands of autonomous domains. Such MOEA/D variants mentioned in this article could be used in one domain to optimize the above performance metrics which are diversionary or in conflict with each other. However, for the whole network, we may need to employ the multi-agent optimization architecture in which each of all MOEA/D optimizers or agents works with other ones together, exploring the global optimization.

### 2) NFV

In traditional networks, network functions are implemented by the dedicated hardware (e.g., video coder/decoder). However, in NFV scenarios, these functions are implemented

in software that runs on top of general-purpose hardware like commercial servers [106], thus flexibly and efficiently utilizing network resources while reducing Capex and Opex. One of the important issues is the resource allocation in NFV (RA-NFV) scenarios. Each user request is described by a service function chain (SFC), and the RA-NFV procedure consists of three stages of SFC constructing, mapping, and scheduling [107], each of which could work independently or collaboratively with up to two others. Much attention has been focused on SFC deployments. Sun *et al.* [108] designed a heuristic algorithm to realize low-latency and resource-efficient SFC orchestration. Similar to the idea of decomposition in MOEA/D, Sun *et al.* [109] split a large flow into a number of subflows and each subflow was redirected to one of these “parallelized” sub-SFCs. Xu *et al.* [110] considered the requested resource quantity of each virtual network function (VNF) and VNFs precedence, which means that, for an SFC, its one VNF must be executed before the special VNF.

To face the challenges resulting from three stages and additional constraints in the RA-NFV, we need to add some mechanisms to enhance MOEA/D so that it can globally optimize the RA-NFV even for the case of static SFC. The first one is to embed a heuristic algorithm in MOEA/D to check the feasibility of a solution by inspecting various constraints like VNFs precedence. In the second one, we could combine the physical characteristic of a network to help MOEA/D select a better path when co-evolving SFC mapping and scheduling jointly, such as several centralities of closeness, betweenness, and degree [111]. For the dynamic SFC case, the computational time is stringent, so we have to greatly reduce the objective space and/or search space by using decomposition [25], [78].

### 3) EON

As one of the most promising candidates for future networking, EONs could provide a finer or huger bandwidth granularity, resulting in high network utilization with a better match between the user request and network resource. The bandwidth unit in EONs is referred to as the frequency slot, and its resource allocation performance includes blocking probability [112], resource utilization, lightpath setup time [113], network throughput and network bandwidth fragmentation ratio [114], and load balancing. The application of EONs to such fields as SDN [115], IT resources [116] and NFV [107] is quite challenging. Facing these different scenarios, we need to improve the performance of MOEA/D comprehensively by employing several methods reviewed above or combining the methods that benefit each other complementally.

In order to achieve a better application effect for the three issues and others, an inevitable trend is to introduce more machine learning tools like deep neuronal networks into MOEA/D, realizing the combination of offline learning and online optimization.

## VI. CONCLUSION

This article provides a relatively comprehensive overview of the research on the improvement and extension of the original MOEA/D in the past decade. It also presents the state-of-the-art MOEA/D with challenges and future research directions. To solve the problems with an increasing number of objectives, constraints and decision variables, it is expected that the satisfactory solutions in different engineering fields could be obtained if our readers could use so many mechanisms, methods and techniques addressed in this survey, by selectively employing some of them, integrating or combining some different kinds of them, or even embedding some cutting techniques like AI-based techniques in MOEA/D.

Many other future research possibilities exist, including developing efficient algorithms that can adapt to more occasions by taking advantages of mixing and parallelism, and a comparison of various MOEA/D improvement methods. Our next study is to extend MOEA/D to other hotspot areas, such as large-scale optimization, edge computing, and machine learning. Besides, it is recommended that researchers should investigate future network optimization using MOEA/D in the aspects of transmission, switching, application and overall network architecture.

## REFERENCES

- [1] Y. Gu, Y. Hu, Y. Ding, J. Lu, and J. Xie, “Elastic virtual network function orchestration policy based on workload prediction,” *IEEE Access*, vol. 7, pp. 96868–96878, 2019.
- [2] A. Trivedi, D. Srinivasan, K. Sanyal, and A. Ghosh, “A survey of multi-objective evolutionary algorithms based on decomposition,” *IEEE Trans. Evol. Comput.*, vol. 21, no. 3, pp. 440–462, Jun. 2017.
- [3] Q. Zhang and H. Li, “MOEA/D: A multiobjective evolutionary algorithm based on decomposition,” *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712–731, Dec. 2007.
- [4] Q. Zhang, B. Li, and F. Zhang, “A MOEA/D approach to exploit the crucial structure of convolution kernels,” in *Proc. 10th Int. Conf. Adv. Comput. Intell. (ICACI)*, Xiamen, China, Mar. 2018, pp. 643–648.
- [5] J. Liu, M. Gong, Q. Miao, X. Wang, and H. Li, “Structure learning for deep neural networks based on multiobjective optimization,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 6, pp. 2450–2463, Jun. 2018.
- [6] M. Kitahashi and H. Handa, “Application of evolutionary multiobjective optimization in L1-regularization of CNN,” in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Wellington, New Zealand, Jun. 2019, pp. 2129–2135.
- [7] S.-L. Jiang and L. Zhang, “Energy-oriented scheduling for hybrid flow shop with limited buffers through efficient multi-objective optimization,” *IEEE Access*, vol. 7, pp. 34477–34487, 2019.
- [8] E. Jiang and L. Wang, “A modified MOEA/D for energy-efficient flexible job shop scheduling problem,” in *Proc. IEEE Int. Conf. Ind. Eng. Manage. (IEMM)*, Bangkok, Thailand, Dec. 2018, pp. 1476–1480.
- [9] Z. Cao, K. Liu, and B. Hu, “An improved MOEA/D for order scheduling problem in automated warehouse,” in *Proc. IEEE 14th Int. Conf. Autom. Sci. Eng. (CASE)*, Munich, Germany, Aug. 2018, pp. 797–802.
- [10] Y. Xu, O. Ding, R. Qu, and K. Li, “Hybrid multi-objective evolutionary algorithms based on decomposition for wireless sensor network coverage optimization,” *Appl. Soft Comput.*, vol. 68, pp. 268–282, Jul. 2018.
- [11] M. Gamal, M. Abolhasan, J. Lipman, R. P. Liu, and W. Ni, “Multi objective resource optimisation for network function virtualisation requests,” in *Proc. 26th Int. Conf. Syst. Eng. (ICSEng)*, Sydney, NSW, Australia, Dec. 2018, pp. 1–7.
- [12] F. Y. Okay and S. Ozdemir, “Efficient and secure coverage control in Internet of Things,” in *Proc. 3rd Int. Conf. Comput. Sci. Eng. (UBMK)*, Sarajevo, Herzegovina, Sep. 2018, pp. 90–95.



- [13] X. Ma, F. Liu, Y. Qi, L. Li, L. Jiao, X. Deng, X. Wang, B. Dong, Z. Hou, Y. Zhang, and J. Wu, "MOEA/D with biased weight adjustment inspired by user preference and its application on multi-objective reservoir flood control problem," *Soft Comput.*, vol. 20, no. 12, pp. 4999–5023, Dec. 2016.
- [14] H. Zhang, D. Yue, W. Yue, K. Li, and M. Yin, "MOEA/D-based probabilistic PBI approach for risk-based optimal operation of hybrid energy system with intermittent power uncertainty," *IEEE Trans. Syst., Man, Cybern., Syst.*, to be published, doi: [10.1109/TSMC.2019.2931636](https://doi.org/10.1109/TSMC.2019.2931636).
- [15] H. Zhang, Y. Zhao, F. Wang, A. Zhang, P. Yang, and X. Shen, "A new evolutionary algorithm based on MOEA/D for portfolio optimization," in *Proc. 10th Int. Conf. Adv. Comput. Intell. (ICACI)*, Xiamen, China, Mar. 2018, pp. 831–836.
- [16] H. Sato, "Analysis of inverted PBI and comparison with other scalarizing functions in decomposition based MOEAs," *J. Heuristics*, vol. 21, no. 6, pp. 819–849, Dec. 2015.
- [17] L. Wang, Q. Zhang, A. Zhou, M. Gong, and L. Jiao, "Constrained subproblems in a decomposition-based multiobjective evolutionary algorithm," *IEEE Trans. Evol. Comput.*, vol. 20, no. 3, pp. 475–480, Jun. 2016.
- [18] X. Ma, Q. Zhang, G. Tian, J. Yang, and Z. Zhu, "On Tchebycheff decomposition approaches for multiobjective evolutionary optimization," *IEEE Trans. Evol. Comput.*, vol. 22, no. 2, pp. 226–244, Apr. 2018.
- [19] S. Jiang, S. Yang, Y. Wang, and X. Liu, "Scalarizing functions in decomposition-based multiobjective evolutionary algorithms," *IEEE Trans. Evol. Comput.*, vol. 22, no. 2, pp. 296–313, Apr. 2018.
- [20] H. Ishibuchi, Y. Sakane, N. Tsukamoto, and Y. Nojima, "Simultaneous use of different scalarizing functions in MOEA/D," in *Proc. 12th Annu. Conf. Genetic Evol. Comput. (GECCO)*, Portland, OR, USA, 2010, pp. 519–526.
- [21] I. Das and J. E. Dennis, "Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems," *SIAM J. Optim.*, vol. 8, no. 3, pp. 631–657, Aug. 1998.
- [22] Q. Zhang, H. Li, D. Maringer, and E. Tsang, "MOEA/D with NBI-style Tchebycheff approach for portfolio management," in *Proc. IEEE Congr. Evol. Comput.*, Barcelona, Spain, Jul. 2010, pp. 1–8.
- [23] H. Ishibuchi, Y. Sakane, N. Tsukamoto, and Y. Nojima, "Adaptation of scalarizing functions in MOEA/D: An adaptive scalarizing function-based multiobjective evolutionary algorithm," in *Proc. 5th Int. Conf. Evol. Multi-Criterion Optim.*, Nantes, France, 2009, pp. 438–452.
- [24] M. Wu, K. Li, S. Kwong, Q. Zhang, and J. Zhang, "Learning to decompose: A paradigm for decomposition-based multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 23, no. 3, pp. 376–390, Jun. 2019.
- [25] H.-L. Liu, F. Gu, and Q. Zhang, "Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems," *IEEE Trans. Evol. Comput.*, vol. 18, no. 3, pp. 450–455, Jun. 2014.
- [26] Y. Zhang, R. Yang, J. Zuo, and X. Jing, "Enhancing MOEA/D with uniform population initialization, weight vector design and adjustment using uniform design," *J. Syst. Eng. Electron.*, vol. 26, no. 5, pp. 1010–1022, Oct. 2015.
- [27] K. Harada, S. Hiwa, and T. Hiroyasu, "Adaptive weight vector assignment method for MOEA/D," in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, Honolulu, HI, USA, Nov. 2017, pp. 1–9.
- [28] S. Jiang, Z. Cai, J. Zhang, and Y.-S. Ong, "Multiobjective optimization by decomposition with Pareto-adaptive weight vectors," in *Proc. 7th Int. Conf. Natural Comput.*, Shanghai, China, Jul. 2011, pp. 1260–1264.
- [29] Y. Qi, X. Ma, F. Liu, L. Jiao, J. Sun, and J. Wu, "MOEA/D with adaptive weight adjustment," *Evol. Comput.*, vol. 22, no. 2, pp. 231–264, Jun. 2014.
- [30] M. Pilat and R. Neruda, "Incorporating user preferences in MOEA/D through the coevolution of weights," in *Proc. Genetic Evol. Comput. Conf. (GECCO)*, Madrid, Spain, 2015, pp. 727–734.
- [31] H. Li, M. Ding, J. Deng, and Q. Zhang, "On the use of random weights in MOEA/D," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Sendai, Japan, May 2015, pp. 978–985.
- [32] H. Li and Q. Zhang, "Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 13, no. 2, pp. 284–302, Apr. 2009.
- [33] T.-C. Chiang and Y.-P. Lai, "MOEA/D-AMS: Improving MOEA/D by an adaptive mating selection mechanism," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, New Orleans, LA, USA, Jun. 2011, pp. 1473–1480.
- [34] S. Jiang and S. Yang, "An improved multiobjective optimization evolutionary algorithm based on decomposition for complex Pareto fronts," *IEEE Trans. Cybern.*, vol. 46, no. 2, pp. 421–437, Feb. 2016.
- [35] Z. Wang, Q. Zhang, M. Gong, and A. Zhou, "A replacement strategy for balancing convergence and diversity in MOEA/D," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Beijing, China, Jul. 2014, pp. 2132–2139.
- [36] Z. Wang, Q. Zhang, A. Zhou, M. Gong, and L. Jiao, "Adaptive replacement strategies for MOEA/D," *IEEE Trans. Cybern.*, vol. 46, no. 2, pp. 474–486, Feb. 2016.
- [37] A. Zamuda, J. Brest, B. Boskovic, and V. Zumer, "Differential evolution with self-adaptation and local search for constrained multiobjective optimization," in *Proc. IEEE Congr. Evol. Comput.*, Trondheim, Norway, May 2009, pp. 195–202.
- [38] L. Ke, Q. Zhang, and R. Battiti, "MOEA/D-ACO: A multiobjective evolutionary algorithm using decomposition and AntColony," *IEEE Trans. Cybern.*, vol. 43, no. 6, pp. 1845–1859, Dec. 2013.
- [39] I. Giagkiozis, R. C. Purshouse, and P. J. Fleming, "Generalized decomposition," in *Proc. 7th Int. Conf. Evol. Multi-Criterion Optim. (EMO)*, Sheffield, U.K., 2013, pp. 428–442.
- [40] Y. Yuan, H. Xu, B. Wang, B. Zhang, and X. Yao, "Balancing convergence and diversity in decomposition-based many-objective optimizers," *IEEE Trans. Evol. Comput.*, vol. 20, no. 2, pp. 180–198, Apr. 2016.
- [41] Y.-Y. Tan, Y.-C. Jiao, H. Li, and X.-K. Wang, "MOEA/D + uniform design: A new version of MOEA/D for optimization problems with many objectives," *Comput. Oper. Res.*, vol. 40, no. 6, pp. 1648–1660, Jun. 2013.
- [42] X. Ma, Y. Qi, L. Li, F. Liu, L. Jiao, and J. Wu, "MOEA/D with uniform decomposition measurement for many-objective problems," *Soft Comput.*, vol. 18, no. 12, pp. 2541–2564, Dec. 2014.
- [43] C. Dai and Y. Wang, "A new decomposition based evolutionary algorithm with uniform designs for many-objective optimization," *Appl. Soft Comput.*, vol. 30, pp. 238–248, May 2015.
- [44] C. Dai and Y. Wang, "A new uniform evolutionary algorithm based on decomposition and CDAS for many-objective optimization," *Knowl.-Based Syst.*, vol. 85, pp. 131–142, Sep. 2015.
- [45] F. Gu and Y.-M. Cheung, "Self-organizing map-based weight design for decomposition-based many-objective evolutionary algorithm," *IEEE Trans. Evol. Comput.*, vol. 22, no. 2, pp. 211–225, Apr. 2018.
- [46] A. Mohammadi, M. N. Omidvar, X. Li, and K. Deb, "Integrating user preferences and decomposition methods for many-objective optimization," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Beijing, China, Jul. 2014, pp. 421–428.
- [47] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff, "A reference vector guided evolutionary algorithm for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 20, no. 5, pp. 773–791, Oct. 2016.
- [48] Y. Chen and A. Zhou, "MOEA/D with an improved multi-dimensional mapping coding scheme for constrained multi-objective portfolio optimization," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Wellington, New Zealand, Jun. 2019, pp. 1742–1749.
- [49] X.-L. Zheng and L. Wang, "A collaborative multiobjective fruit fly optimization algorithm for the resource constrained unrelated parallel machine green scheduling problem," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 48, no. 5, pp. 790–800, May 2018.
- [50] N. G. Paterakis, M. Gibescu, A. G. Bakirtzis, and J. P. S. Catalao, "A multi-objective optimization approach to risk-constrained energy and reserve procurement using demand response," *IEEE Trans. Power Syst.*, vol. 33, no. 4, pp. 3940–3954, Jul. 2018.
- [51] M. A. Jan and Q. Zhang, "MOEA/D for constrained multiobjective optimization: Some preliminary experimental results," in *Proc. UK Workshop Comput. Intell. (UKCI)*, Colchester, U.K., Sep. 2010, pp. 1–6.
- [52] Z. Fan, W. Li, X. Cai, H. Li, C. Wei, Q. Zhang, K. Deb, and E. Goodman, "Push and pull search for solving constrained multi-objective optimization problems," *Swarm Evol. Comput.*, vol. 44, pp. 665–679, Feb. 2019.
- [53] Z. Fan, J. Ruan, W. Li, Y. You, X. Cai, Z. Xu, Z. Yang, F. Sun, Z. Wang, Y. Yuan, Z. Li, and G. Zhu, "A learning guided parameter setting for constrained multi-objective optimization," in *Proc. 1st Int. Conf. Ind. Artif. Intell. (IAI)*, Shenyang, China, Jul. 2019, pp. 1–6.
- [54] Z. Fan, H. Li, C. Wei, W. Li, H. Huang, X. Cai, and Z. Cai, "An improved epsilon constraint handling method embedded in MOEA/D for constrained multi-objective optimization problems," in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, Athens, Greece, Dec. 2016, pp. 1–8.
- [55] Z. Fan, Y. Fang, W. Li, X. Cai, C. Wei, and E. Goodman, "MOEA/D with angle-based constrained dominance principle for constrained multi-objective optimization problems," *Appl. Soft Comput.*, vol. 74, pp. 621–633, Jan. 2019.

- [56] Z. Fan, Y. Fang, W. Li, J. Lu, X. Cai, and C. Wei, "A comparative study of constrained multi-objective evolutionary algorithms on constrained multi-objective optimization problems," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Donostia-San Sebastian, Spain, Jun. 2017, pp. 209–216.
- [57] Z. Fan, H. Huang, W. Li, S. Xie, X. Cai, and E. Goodman, "An opposition-based repair operator for multi-objective evolutionary algorithm in constrained optimization problems," in *Proc. 11th Int. Conf. Natural Comput. (ICNC)*, Zhangjiajie, China, Aug. 2015, pp. 330–336.
- [58] X. Ma, F. Liu, Y. Qi, M. Gong, M. Yin, L. Li, L. Jiao, and J. Wu, "MOEA/D with opposition-based learning for multiobjective optimization problem," *Neurocomputing*, vol. 146, pp. 48–64, Dec. 2014.
- [59] M.-Y. Ameca-Alducin, M. Hasani-Shoreh, and F. Neumann, "On the use of repair methods in differential evolution for dynamic constrained optimization," in *Applications of Evolutionary Computation* (Lecture Notes in Computer Science), vol. 10784, K. Sim and P. Kaufmann, Eds. Cham, Switzerland: Springer, 2018, pp. 832–847.
- [60] S. Yang, S. Jiang, and Y. Jiang, "Improving the multiobjective evolutionary algorithm based on decomposition with new penalty schemes," *Soft Comput.*, vol. 21, no. 16, pp. 4677–4691, Aug. 2017.
- [61] M. Ming, R. Wang, Y. Zha, and T. Zhang, "Pareto adaptive penalty-based boundary intersection method for multi-objective optimization," *Inf. Sci.*, vol. 414, pp. 158–174, Nov. 2017.
- [62] X. Cai, Z. Mei, Z. Fan, and Q. Zhang, "A constrained decomposition approach with grids for evolutionary multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 22, no. 4, pp. 564–577, Aug. 2018.
- [63] W. Zheng, Y. Tan, L. Meng, and H. Zhang, "An improved MOEA/D design for many-objective optimization problems," *Appl. Intell.*, vol. 48, no. 10, pp. 3839–3861, Oct. 2018.
- [64] R. Wang, Q. Zhang, and T. Zhang, "Decomposition-based algorithms using Pareto adaptive scalarizing methods," *IEEE Trans. Evol. Comput.*, vol. 20, no. 6, pp. 821–837, Dec. 2016.
- [65] Y.-Y. Tan, Y.-C. Jiao, and X.-K. Wang, "MOEA/D with uniform design for the multiobjective 0/1 knapsack problem," in *Proc. 7th Int. Conf. Comput. Intell. Secur.*, Hainan, China, Dec. 2011, pp. 96–100.
- [66] Z. Li, L. He, and Y. Chu, "An improved decomposition multiobjective optimization algorithm with weight vector adaptation strategy," in *Proc. 13th Int. Conf. Semantics, Knowl. Grids (SKG)*, Beijing, China, Aug. 2017, pp. 19–24.
- [67] Y. Qi, X. Ma, M. Yin, F. Liu, and J. Wei, "MOEA/D with a delaunay triangulation based weight adjustment," in *Proc. Conf. Companion Genetic Evol. Comput. Companion (GECCO)*, Vancouver, BC, Canada, 2014, pp. 93–94.
- [68] H. Xu, W. Zeng, D. Zhang, and X. Zeng, "MOEA/HD: A multiobjective evolutionary algorithm based on hierarchical decomposition," *IEEE Trans. Cybern.*, vol. 49, no. 2, pp. 517–526, Feb. 2019.
- [69] K. Li, Q. Zhang, S. Kwong, M. Li, and R. Wang, "Stable matching-based selection in evolutionary multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 18, no. 6, pp. 909–923, Dec. 2014.
- [70] H. Ishibuchi, T. Doi, and Y. Nojima, "Effects of using two neighborhood structures in cellular genetic algorithms for function optimization," in *Proc. 9th Int. Conf. Parallel Problem Solving Nature*, Reykjavik, Iceland, 2006, pp. 949–958.
- [71] S.-Z. Zhao, P. N. Suganthan, and Q. Zhang, "Decomposition-based multiobjective evolutionary algorithm with an ensemble of neighborhood sizes," *IEEE Trans. Evol. Comput.*, vol. 16, no. 3, pp. 442–446, Jun. 2012.
- [72] K. Li, A. Fialho, S. Kwong, and Q. Zhang, "Adaptive operator selection with bandits for a multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 18, no. 1, pp. 114–130, Feb. 2014.
- [73] X. Ma, F. Liu, Y. Qi, L. Li, L. Jiao, M. Liu, and J. Wu, "MOEA/D with Baldwinian learning inspired by the regularity property of continuous multiobjective problem," *Neurocomputing*, vol. 145, pp. 336–352, Dec. 2014.
- [74] Z. Wang, Q. Zhang, and H. Li, "Balancing convergence and diversity by using two different reproduction operators in MOEA/D: Some preliminary work," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Hong Kong, Oct. 2015, pp. 2849–2854.
- [75] H. Li, Q. Zhang, and J. Deng, "Biased multiobjective optimization and decomposition algorithm," *IEEE Trans. Cybern.*, vol. 47, no. 1, pp. 52–66, Jan. 2017.
- [76] I. Giagkiozis, R. C. Purshouse, and P. J. Fleming, "Generalized decomposition and cross entropy methods for many-objective optimization," *Inf. Sci.*, vol. 282, pp. 363–387, Oct. 2014.
- [77] M. Asafuddoula, T. Ray, and R. Sarker, "A decomposition-based evolutionary algorithm for many objective optimization," *IEEE Trans. Evol. Comput.*, vol. 19, no. 3, pp. 445–460, Jun. 2015.
- [78] H.-L. Liu, L. Chen, Q. Zhang, and K. Deb, "An evolutionary many-objective optimisation algorithm with adaptive region decomposition," in *Proc. 2016 IEEE Congr. Evol. Comput. (CEC)*, Vancouver, BC, Canada, Jul. 2016, pp. 4763–4769.
- [79] L. Chen, H.-L. Liu, and K. C. Tan, "Decomposition based dominance relationship for evolutionary many-objective algorithm," in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, Honolulu, HI, USA, Nov. 2017, pp. 1–6.
- [80] Z. Lin, H. Liu, and F. Gu, "An evolutionary multi- and many-objective optimization algorithm based on ISDE + and region decomposition," in *Proc. 14th Int. Conf. Comput. Intell. Secur. (CIS)*, Hangzhou, China, Nov. 2018, pp. 30–34.
- [81] K. Li, K. Deb, Q. Zhang, and S. Kwong, "An evolutionary many-objective optimization algorithm based on dominance and decomposition," *IEEE Trans. Evol. Comput.*, vol. 19, no. 5, pp. 694–716, Oct. 2015.
- [82] L. Chen, H.-L. Liu, K. C. Tan, Y.-M. Cheung, and Y. Wang, "Evolutionary many-objective algorithm using decomposition-based dominance relationship," *IEEE Trans. Cybern.*, vol. 49, no. 12, pp. 4129–4139, Dec. 2019.
- [83] H. Ishibuchi, Y. Setoguchi, H. Masuda, and Y. Nojima, "Performance of decomposition-based many-objective algorithms strongly depends on Pareto front shapes," *IEEE Trans. Evol. Comput.*, vol. 21, no. 2, pp. 169–190, Apr. 2017.
- [84] X. He, Y. Zhou, Z. Chen, and Q. Zhang, "Evolutionary many-objective optimization based on dynamical decomposition," *IEEE Trans. Evol. Comput.*, vol. 23, no. 3, pp. 361–375, Jun. 2019.
- [85] K. Li, R. Chen, G. Min, and X. Yao, "Integration of preferences in decomposition multiobjective optimization," *IEEE Trans. Cybern.*, vol. 48, no. 12, pp. 3359–3370, Dec. 2018.
- [86] L. Li, H. Chen, J. Li, N. Jing, and M. Emmerich, "Integrating region preferences in multiobjective evolutionary algorithms based on decomposition," in *Proc. 10th Int. Conf. Adv. Comput. Intell. (ICACI)*, Xiamen, China, Mar. 2018, pp. 379–384.
- [87] M. Xiong, W. Xiong, and C. Liu, "A hybrid many-objective evolutionary algorithm with region preference for decision makers," *IEEE Access*, vol. 7, pp. 117699–117715, 2019.
- [88] Z. He and G. G. Yen, "Diversity improvement in decomposition-based multi-objective evolutionary algorithm for many-objective optimization problems," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, San Diego, CA, USA, Oct. 2014, pp. 2409–2414.
- [89] X. Cai, Z. Mei, and Z. Fan, "A decomposition-based many-objective evolutionary algorithm with two types of adjustments for direction vectors," *IEEE Trans. Cybern.*, vol. 48, no. 8, pp. 2335–2348, Aug. 2018.
- [90] H. Sato, S. Nakagawa, M. Miyakawa, and K. Takadama, "Enhanced decomposition-based many-objective optimization using supplemental weight vectors," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Vancouver, BC, Canada, Jul. 2016, pp. 1626–1633.
- [91] R. Kia, P. Shahnazari-Shahrezaei, and S. Zabihi, "Solving a multi-objective mathematical model for a multi-skilled project scheduling problem by CPLEX solver," in *Proc. IEEE Int. Conf. Ind. Eng. Eng. Manage. (IEEM)*, Bali, IN, USA, Dec. 2016, pp. 1220–1224.
- [92] A. D. Haidar, "Operations research and optimization techniques," in *Construction Program Management—Decision Making and Optimization Techniques*, 1st ed. Cham, Switzerland: Springer, 2016, ch. 5, pp. 131–157.
- [93] F. Gu, L. Han, M. Zheng, H.-L. Liu, and X. Chen, "Local search based constrained evolutionary multiobjective algorithm for objective reduction," in *Proc. 9th Int. Conf. Inf. Sci. Technol. (ICIST)*, Hulunbuir, China, Aug. 2019, pp. 169–174.
- [94] Y. Yuan, Y.-S. Ong, A. Gupta, and H. Xu, "Objective reduction in many-objective optimization: Evolutionary multiobjective approaches and comprehensive analysis," *IEEE Trans. Evol. Comput.*, vol. 22, no. 2, pp. 189–210, Apr. 2018.
- [95] X. Ma, F. Liu, Y. Qi, X. Wang, L. Li, L. Jiao, M. Yin, and M. Gong, "A multiobjective evolutionary algorithm based on decision variable analyses for multiobjective optimization problems with large-scale variables," *IEEE Trans. Evol. Comput.*, vol. 20, no. 2, pp. 275–298, Apr. 2016.

- [96] A. Ibrahim, M. V. Martin, S. Rahnamayan, and K. Deb, "Fusion-based hybrid many-objective optimization algorithm," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Donostia-San Sebastian, Spain, Jun. 2017, pp. 2372–2381.
- [97] X. Zhang, Y. Tian, R. Cheng, and Y. Jin, "A decision variable clustering-based evolutionary algorithm for large-scale many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 22, no. 1, pp. 97–112, Feb. 2018.
- [98] X. Wu, Y. Wang, J. Liu, and N. Fan, "A new hybrid algorithm for solving large scale global optimization problems," *IEEE Access*, vol. 7, pp. 103354–103364, 2019.
- [99] C. He, L. Li, Y. Tian, X. Zhang, R. Cheng, Y. Jin, and X. Yao, "Accelerating large-scale multiobjective optimization via problem reformulation," *IEEE Trans. Evol. Comput.*, vol. 23, no. 6, pp. 949–961, Dec. 2019.
- [100] B. Cao, J. Zhao, Z. Lv, and X. Liu, "A distributed parallel cooperative coevolutionary multiobjective evolutionary algorithm for large-scale optimization," *IEEE Trans. Ind. Informat.*, vol. 13, no. 4, pp. 2030–2038, Aug. 2017.
- [101] Y.-H. Jia, Y.-R. Zhou, Y. Lin, W.-J. Yu, Y. Gao, and L. Lu, "A distributed cooperative co-evolutionary CMA evolution strategy for global optimization of large-scale overlapping problems," *IEEE Access*, vol. 7, pp. 19821–19834, 2019.
- [102] Y. Zhao, W. Wang, Y. Li, C. Colman Meixner, M. Tornatore, and J. Zhang, "Edge computing and networking: A survey on infrastructures and applications," *IEEE Access*, vol. 7, pp. 101213–101230, 2019.
- [103] X. Huang, S. Cheng, K. Cao, P. Cong, T. Wei, and S. Hu, "A survey of deployment solutions and optimization strategies for hybrid SDN networks," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 2, pp. 1483–1507, 2nd Quart., 2019.
- [104] S. Ejaz, Z. Iqbal, P. A. Shah, B. H. Bukhari, A. Ali, and F. Aadil, "Traffic load balancing using software defined networking (SDN) controller as virtualized network function," *IEEE Access*, vol. 7, pp. 46646–46658, 2019.
- [105] S. Garg, K. Kaur, S. H. Ahmed, A. Bradai, G. Kaddoum, and M. Atiquzzaman, "MobQoS: Mobility-aware and QoS-driven SDN framework for autonomous vehicles," *IEEE Wireless Commun.*, vol. 26, no. 4, pp. 12–20, Aug. 2019.
- [106] J. Gil Herrera and J. F. Botero, "Resource allocation in NFV: A comprehensive survey," *IEEE Trans. Netw. Serv. Manage.*, vol. 13, no. 3, pp. 518–532, Sep. 2016.
- [107] G. Mirjalili and Z. Luo, "Optimal network function virtualization and service function chaining: A survey," *Chin. J. Electron.*, vol. 27, no. 4, pp. 704–717, Jul. 2018.
- [108] G. Sun, Z. Xu, H. Yu, X. Chen, V. Chang, and A. V. Vasilakos, "Low-latency and resource-efficient service function chaining orchestration in network function virtualization," *IEEE Internet Things J.*, to be published, doi: [10.1109/JIOT.2019.2937110](https://doi.org/10.1109/JIOT.2019.2937110).
- [109] G. Sun, Z. Chen, H. Yu, X. Du, and M. Guizani, "Online parallelized service function chain orchestration in data center networks," *IEEE Access*, vol. 7, pp. 100147–100161, 2019.
- [110] Z. Xu, C. Jin, F. Yang, Z. Ding, S. Du, and G. Zhao, "Coordinated resource allocation with VNFs precedence constraints in inter-datacenter networks over elastic optical infrastructure," in *Proc. IEEE World Symp. Commun. Eng. (WSCE)*, Singapore, Dec. 2018, pp. 1–6.
- [111] M. E. J. Newman, "Networks: An introduction," *Astron. Nachr.*, vol. 327, no. 8, pp. 741–743, Jan. 2010.
- [112] Y. Zhao, L. Hu, R. Zhu, X. Yu, X. Wang, and J. Zhang, "Crosstalk-aware spectrum defragmentation based on spectrum compactness in space division multiplexing enabled elastic optical networks with multicore fiber," *IEEE Access*, vol. 6, pp. 15346–15355, 2018.
- [113] A. S. Thyagaturu, A. Mercian, M. P. McGarry, M. Reisslein, and W. Kellerer, "Software defined optical networks (SDONs): A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 4, pp. 2738–2786, 4th Quart., 2016.
- [114] Z. Zhu, W. Lu, L. Zhang, and N. Ansari, "Dynamic service provisioning in elastic optical networks with hybrid single-/multi-path routing," *J. Lightw. Technol.*, vol. 31, no. 1, pp. 15–22, Jan. 2013.
- [115] Y. Xiong, J. Shi, Y. Yang, Y. Lv, and G. N. Rouskas, "Lightpath management in SDN-based elastic optical networks with power consumption considerations," *J. Lightw. Technol.*, vol. 36, no. 9, pp. 1650–1660, May 1, 2018.
- [116] B. Li, X. Yu, Y. Zhao, Y. Li, Q. Ou, Z. Liu, X. Liao, and J. Zhang, "Joint IT and spectrum resource load balancing for VNF selection in inter-datacenter elastic optical networks," in *Proc. 16th Int. Conf. Opt. Commun. Netw. (ICOON)*, Wuzhen, China, Aug. 2017, pp. 1–3.



**QIAN XU** received the B.S. degree in communication engineering from Chang'an University, Xi'an, China, in 2017. She is currently pursuing the M.S. degree with the State Key Laboratory of Integrated Services Networks (ISN), Xidian University, Xi'an. Her current research interests are in the areas of elastic optical networks and multi-objective optimization.



**ZHANQI XU** (Senior Member, IEEE) received the B.S., M.S., and Ph.D. degrees in communication and electronic system from Xidian University, Xi'an, China, in 1984, 1987, and 1997, respectively. He had a one-year postdoctoral study at Hong Kong University of Science and Technology, during the turn of this century. Since 2000, he has been with the State Key Laboratory of Integrated Services Networks (ISN), Xidian University, China, where he is currently a Professor.

He has authored 40 international journal and conference papers. His research interests include novel network architecture like SDN/NFV, communication networks optimization modeling, and intelligent algorithm, optical networks, and space information networks. He is a Senior Member of China Institute of Communication (CIC) and China Computer Federation (CCF).



**TAO MA** (Member, IEEE) received the B.Sc. and M.Sc. degrees in electrical engineering from Xi'an Jiaotong University, Xi'an, China, in 2005 and 2008, respectively, and the Ph.D. degree from the University of Nebraska-Lincoln, NE, USA. He is currently an Associate Professor with the State Key Laboratory of Integrated Services Networks (ISN), Xidian University, China. His research areas are the IoT architecture, backscatter communication, and batteryless communication system design.

• • •