Multiobjectivization of Single-Objective Optimization in Evolutionary Computation: A Survey

Xiaoliang Ma, *Member, IEEE*, Zhitao Huang, Xiaodong Li, *Fellow, IEEE*, Yutao Qi, Lei Wang, and Zexuan Zhu, *Senior Member, IEEE*

***Abstract*—Multiobjectivization has emerged as a new promising paradigm to solve single-objective optimization problems (SOPs) in evolutionary computation, where an SOP is transformed into a multiobjective optimization problem (MOP) and solved by an evolutionary algorithm to find the optimal solutions of the original SOP. The transformation of an SOP into an MOP can be done by adding helper-objective(s) into the original objective, decomposing the original objective into multiple subobjectives, or aggregating subobjectives of the original objective into multiple scalar objectives. Multiobjectivization bridges the gap between SOPs and MOPs by transforming an SOP into the counterpart MOP, through which multiobjective optimization methods manage to attain superior solutions of the original SOP. Particularly, using multiobjectivization to solve SOPs can reduce the number of local optima, create new search paths from local optima to global optima, attain more incomparability solutions, and/or improve solution diversity. Since the term "multiobjectivization" was coined by Knowles *et al.* in 2001, this subject has accumulated plenty of works in the last two decades, yet there is a lack of systematic and comprehensive survey of these efforts. This article presents a comprehensive multifacet survey of the state-of-the-art multiobjectivization methods. Particularly, a new taxonomy of the methods is provided in this article and the advantages, limitations, challenges, theoretical analyses, benchmarks, applications, as well as future directions of the multiobjectivization methods are discussed.**

***Index Terms*—Multiobjectivization via decomposition (MVD), multiobjectivization via helper-objectives (MHOs), multiobjectivization via scalarizing functions (MSFs), multiobjectivization.**

Manuscript received May 25, 2021; revised August 4, 2021; accepted October 12, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant 61976143, Grant 61871272, Grant 61772392, and Grant 61975135; in part by the International Cooperation and Exchanges NSFC under Grant 61911530218; in part by the Guangdong Basic and Applied Basic Research Foundation under Grant 2019A1515010869, Grant 2020A1515010946, and Grant 2021A1515012637; in part by the Shenzhen Fundamental Research Program under Grant JCYJ20190808173617147; in part by the Guangdong Provincial Key Laboratory under Grant 2020B121201001; in part by the Scientific Research Foundation of Shenzhen University for Newly Introduced Teachers under Grant 860/2110312; in part by the Science Basic Research Plan in Shaanxi Province of China under Grant 2018JM6009; and in part by the BGI-Research Shenzhen Open Funds under Grant BGIRSZ20200002. This article was recommended by Associate Editor G.-P. Liu. *(Corresponding author: Zexuan Zhu.)*

Xiaoliang Ma and Zhitao Huang are with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China (e-mail: maxiaoliang@szu.edu.cn; huangzhitao@email.szu.edu.cn).

Xiaodong Li is with the School of Science (Computer Science and Software Engineering), RMIT University, Melbourne, VIC 3001, Australia (e-mail: xiaodong.li@rmit.edu.au).

Yutao Qi is with the School of Computer Science and Technology, Xidian University, Xi'an 710071, China (e-mail: ytqi@xidian.edu.cn).

Lei Wang is with the Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China (e-mail: lei.wang1@siat.ac.cn).

Zexuan Zhu is with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China, also with the Shenzhen Pengcheng Laboratory, Shenzhen 518055, China, and also with the Guangdong Provincial Key Laboratory of Brain-Inspired Intelligent Computation, Southern University of Science and Technology, Shenzhen 518055, China (e-mail: zhuzx@szu.edu.cn).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TCYB.2021.3120788.

Digital Object Identifier 10.1109/TCYB.2021.3120788

I. Introduction

The term "multiobjectivization" in evolutionary computation initially refers to the process of reformulating an single-objective optimization problem (SOP) into an multiobjective optimization problem (MOP) that can be solved by a multiobjective optimization method to find the optimal solution(s) of the original SOP [1], [2]. To date, multiobjectivization is normally achieved by adding help-objectives to the original SOP [3], [4], decomposing the original objective into multiple subobjectives [1], [5], or aggregating subobjectives of the original objective into multiple scalar objectives [6]. Solving the reformulated MOP instead of the original SOP can lead to a better solution of the original SOP as the multiobjectivization can reduce the number of local optima [1], [7], introduce more plateaus of incomparability solutions [7], [8], and/or improve the population diversity [1]–[3] (different from the diversity-based methods considering diversity as an objective, the multiobjectivization methods aim at transforming an SOP into an MOP by transforming its fitness landscape [1]). Moreover, multiobjectivization is more informative for leveraging domain knowledge or problem structure of the original SOP [9], [10]. The advantage of multiobjectivization is more obvious in complex SOPs of multimodality, deception, epistasis, and neutrality features that hamper the single-objective evolutionary algorithms (SOEAs) in finding the optimal solutions [1], [3], [11]. These features cause difficulties for single-criterion optimization to determine which solutions to breed and survive, but that are less challenging for multicriterion optimization [2].

TABLE I
ILLUSTRATION OF THE THREE TYPES OF MULTIOBJECTIVIZATION METHODS ON A SUM-OF-PARTS PROBLEM

Original problem	Basic idea	A toy example $f(\mathbf{x}) = f_1(\mathbf{x}) + f_2(\mathbf{x}) + f_3(\mathbf{x})$
Multi-objectivization via decomposition	The original SOP $f(\mathbf{x})$ is decomposed into multiple smaller subproblems that form an MOP	$[f_1(\mathbf{x}), f_2(\mathbf{x}), f_3(\mathbf{x})]^T$, $[f_1(\mathbf{x}), f_2(\mathbf{x}) + f_3(\mathbf{x})]^T$, $[f_2(\mathbf{x}), f_1(\mathbf{x}) + f_3(\mathbf{x})]^T$ $[f_3(\mathbf{x}), f_1(\mathbf{x}) + f_2(\mathbf{x})]^T$
Multi-objectivization via helper-objectives	Helper-objectives are added together with the original objective(s) to form the MOP	$[f(\mathbf{x}), f_1(\mathbf{x})]^T$, $[f(\mathbf{x}), f_2(\mathbf{x})]^T$, $[f(\mathbf{x}), f_3(\mathbf{x})]^T$, $[f(\mathbf{x}), f_1(\mathbf{x}), f_2(\mathbf{x})]^T$, or $[f(\mathbf{x}), f_1(\mathbf{x}), f_2(\mathbf{x}), f_3(\mathbf{x})]^T$
Multi-objectivization via scalarizing functions	Multiple weight vectors are introduced to transform a SOP/MOP into an MOP	$[\sum_{i=1}^{3} w_i \cdot f_i(\mathbf{x}), \ldots, \sum_{i=1}^{3} u_i \cdot f_i(\mathbf{x})]^T$

where v, w_i, and $u_i \in [0, 1]$, $\sum_{i=1}^{3} w_i = 1$ and $\sum_{i=1}^{3} u_i = 1$.

The purpose of this work is to present a comprehensive survey of the state-of-the-art multiobjectivization methods in evolutionary computation. The focus of this survey is on how to transform an SOP into an MOP rather than the evolutionary algorithms (EAs) designed for solving the SOP/MOP. Based on the way the target MOP is constructed, the multiobjectivization methods can be categorized into three groups, that is: 1) multiobjectivization via decomposition (MVD); 2) multiobjectivization via helper-objectives (MHOs); and 3) multiobjectivization via scalarizing functions (MSFs). To give an intuitive view of the three types of multiobjectivization methods, an illustration is provided in Table I, where the original SOP is $f(\mathbf{x}) = f_1(\mathbf{x}) + f_2(\mathbf{x}) + f_3(\mathbf{x})$.

The first type of multiobjectivization method is MVD that was first proposed in [1]. In MVD, the original SOP is decomposed into multiple smaller subproblems that form an MOP. The subproblems should cover all portions of the original SOP so that solving the subproblems simultaneously can lead to the complete solution of the original problem [12]. Moreover, the Pareto set (PS) of the constructed MOP should include at least one optimal solution to the original problem [13]. Note that an SOP can be decomposed into multiple MOPs. The example in Table I (the 2nd row) shows four alternative decompositions of the original SOP $f(\mathbf{x}) = f_1(\mathbf{x}) + f_2(\mathbf{x}) + f_3(\mathbf{x})$. The decomposed subproblems could be mutual exclusive like $[f_1(\mathbf{x}), f_2(\mathbf{x}), f_3(\mathbf{x})]^T$, $[f_1(\mathbf{x}), f_2(\mathbf{x}) + f_3(\mathbf{x})]^T$, $[f_2(\mathbf{x}), f_1(\mathbf{x}) + f_3(\mathbf{x})]^T$, and $[f_3(\mathbf{x}), f_1(\mathbf{x}) + f_2(\mathbf{x})]^T$ or share some overlaps like $[f_1(\mathbf{x}) + v \cdot f_3(\mathbf{x}), f_2(\mathbf{x}) + (1 - v) \cdot f_3(\mathbf{x})]^T$. MVD with overlapping subproblems is also referred to as multiobjectivization via segmentation (MOS) [3], [14]. The choice of decomposition is critical to the performance of multiobjectivization. Smart switch among these decompositions can benefit the search, as the structural changes of the search space help avoid local optima especially on complex problems [15].

The second type of multiobjectivization method is the MHO method, which is also the most widely used type in the literature. MHO adds helper-objectives together with the original objective(s) to form the MOP. The original SOP and the additive helper-objectives are solved simultaneously with the expectation of achieving a better result than merely focusing on the original SOP [16]. As shown in the 3rd row of Table I, five different resultant MOPs are obtained by adding different component helper-objectives to the original SOP. MHO

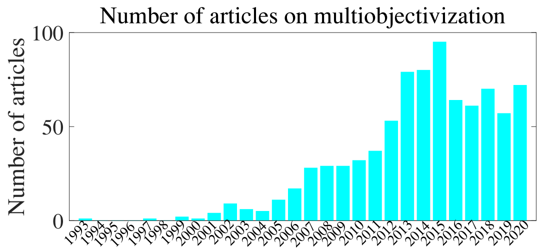


Fig. 1. Numbers of yearly published articles with the term "multiobjectivization" in Google scholar.

can enhance the exploration ability of the EAs by introducing more plateaus of incomparable solutions [8]. Moreover, introducing the helper-objectives can reduce the selection pressure toward the original objective [17] and the search can move through the inferior parts of the original objective space with the help of the helper-objectives. The helper-objectives also provide an opportunity to impose domain knowledge and the preference of the user into the search [3], [18].

The last category MSF introduces multiple weight vectors to transform an SOP/MOP into an MOP [6]. MSF was initially proposed to transform a sum-of-parts problem like $f(\mathbf{x}) = f_1(\mathbf{x}) + \cdots + f_m(\mathbf{x})$ into an MOP by multiple scalarizing functions, for example, $[\sum_{i=1}^{m} w_i \cdot f_i(\mathbf{x}), \ldots, \sum_{i=1}^{m} u_i \cdot f_i(\mathbf{x})]^T$ [6], [19]. Multiple prespecified weight vectors in MSF can serve as the preferences of the decision maker (DM) that enable the construction of a well-designed MOP by specifying weight vectors. In addition, introducing multiple scalarizing functions is to give more freedom of the EAs to alternately explore each objective space of the transformed MOP, and decrease the possibility of getting trapped in local optima [18].

In MVD, the individual component objective of the transformed MOP covers only a portion of the original SOP. MSF differs from MVD in that each component objective in the transformed MOP of MSF involves all parts of the original SOP. MHO can be easily distinguished from MVD and MSF for involving the original SOP as one objective in the final MOP, that is, the original SOP does not directly present in the transformed MOPs of MVD and MSF as shown in Table I.

As the complexity of real-world optimization problems grows rapidly, multiobjectivization has attracted increasing attention by injecting new vitality into the classic optimization solutions [1], [3], [8], [15], [18]–[20]. The number of research works on multiobjectivization boosts in the last two decades (see the yearly published articles on multiobjectivization in Fig. 1). An earlier survey work on multiobjectivization is available in [21], yet a more comprehensive review to reflect the latest development of multiobjectivization is greatly demanded due to rapid advance in this field. The contributions of this work are summarized as follows.

1) This work is the latest systematic survey on multiobjectivization. It gives a new taxonomy of the existing multiobjectivization methods from MVD, MHO, and MSF to static, random, and adaptive methods.
2) This article for the first time subdivides MHO methods into four categories, where the helper-objective is a variation of the original objective $f(\mathbf{x})$, a subobjective of

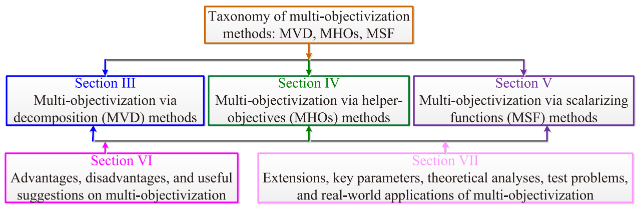

Taxonomy of multi-objectivization methods: MVD, MHOs, MSF
Section III
Multi-objectivization via decomposition (MVD) methods
Section IV
Multi-objectivization via helper-objectives (MHOs) methods
Section V
Multi-objectivization via scalarizing functions (MSF) methods
Section VI
Advantages, disadvantages, and useful suggestions on multi-objectivization
Section VII
Extensions, key parameters, theoretical analyses, test problems, and real-world applications of multi-objectivization

Fig. 2. Connection between Sections III–VIII.

$f(\mathbf{x}), f(\mathbf{x})$ with a different value of the key parameter, or an auxiliary objective without a direct relation with $f(\mathbf{x})$.

3) We offer more in-depth suggestions and reviews of multiobjectivization on the advantages, limitations, useful suggestions, theoretical analysis, key parameters, benchmarks, and applications.
4) New emerging challenges and important future directions of multiobjectivization are summarized. The survey is expected to attract more attention to multiobjectivization and inspire new research ideas and methods.

The remainder of this article is organized as follows. Section II introduces the background. Sections III–V review the three kinds of multiobjectivization methods. Useful suggestions, advantages, and disadvantages of multiobjectivization are discussed in Section VI. Section VII presents the extensions, key parameters, theoretical analyses, test problems, and applications of multiobjectivization. The connection between Sections III–VII is presented in Fig. 2. Finally, we conclude this survey and discuss the potential future directions of multiobjectivization. For the reader's convenience, the nomenclature used in the following text is also provided here.

II. Background

In this section, we introduce the most basic background of the optimization problem and multiobjective EAs to facilitate the understanding the multiobjectivization. More background knowledge is provided in the supplementary material due to the page limit.

A. Single-Objective Optimization Problem and Multiobjective Optimization Problem

Without loss of generality, an optimization problem can be formulated as follows [22], [23]:

$$\begin{cases} \min \mathbf{F}(\mathbf{x}) = \left[g_1(\mathbf{x}), \ldots, g_l(\mathbf{x})\right]^T \\ \text{subject to} : \mathbf{x} \in \Theta \end{cases} \quad (1)$$

where $\Theta \subseteq \mathbf{R}^n$ denotes the decision space, $\mathbf{x} \in \Theta$ is a decision vector of n variables, and $\mathbf{F}(\mathbf{x}) : \Theta \rightarrow \mathbf{R}^l$ indicates an k-element objective function vector. Here, if $l = 1$, the problem is referred to as an SOP; otherwise, the problem is a MOP with $l > 1$. Let $\mathbf{x}, \mathbf{y} \in \Theta$ be two candidate solutions of the problem, $\mathbf{x}$ is said to dominate $\mathbf{y}$ (recorded as $\mathbf{y} \prec \mathbf{x}$), if and only if $g_i(\mathbf{x}) \leq g_i(\mathbf{y})$ for any $i \in \{1, \ldots, l\}$ and $\mathbf{F}(\mathbf{x}) \neq \mathbf{F}(\mathbf{y})$. For an MOP, a solution $\mathbf{x}^* \in \Theta$ is called a Pareto-optimal solution if $\mathbf{x}^*$ is not dominated by any other feasible solution. The set of Pareto-optimal solutions is called the PS and the mapping of PS in objective space is called the Pareto front (PF) [17].

B. Multiobjective Evolutionary Algorithms

As one kind of the mainstream methods for MOPs, multiobjective EAs (MOEAs) can be roughly classified into three categories, that is, dominance-based, decomposition-based, and indicator-based methods. Dominance-based MOEAs have been widely used to solve MOPs and the representative methods include a fast nondominated sorted genetic algorithm (NSGA-II) [23], [24], an improved strength Pareto EA (SPEA2) [25], [26], and a Pareto envelope-based selection algorithm (PESA) [23], [27]). Decomposition-based MOEAs decompose an MOP into multiple single-objective subproblems based on a set of representative weight vectors, and then optimize these subproblems simultaneously in an evolutionary framework. The most well-known methods include the MOEA based on decomposition (MOEA/D), vector-evaluated genetic algorithm (VEGA) [22], [28], and NSGA-III [17]. Indicator-based methods take the indicator value of each individual as the selection criterion. The most widely used indicator-based MOEAs include the indicator selection-based evolution algorithm (IBEA) [25], multiobjective selection based on dominated hypervolume (SMS-EMOA) [29], and hypervolume-based evolutionary many-objective optimization (HypE) [25].

There are many survey papers on EAs designed for MOPs [22], [23], [25], [30] and/or SOPs [31]–[33]. Yet, they normally do not cover the topics on multiobjectivization, which motivates us to provide a comprehensive and multifacet survey on the latest development of multiobjectivization methods.

III. Multiobjectivization via Decomposition

MVD decomposes an SOP into several smaller and easier subproblems and solves all subproblems simultaneously as an MOP to find the optimal solutions of the original SOP. Ideally, problem decomposition should follow the principle of "mutually exclusive, collectively exhaustive," that is, the subproblems should be as independent as possible [1] and each component of the SOP should be covered in the constructed MOP [7].

The existing MVD methods can be classified into static and dynamic decomposition methods. During the process of optimization, the problem decomposition is fixed for static MVD methods and is variable for dynamic MVD methods. Static decomposition methods are simple yet bias the search. Dynamic decomposition methods are less biased as they can dynamically adjust the problem decomposition during the evolution.

A. Static Decomposition Methods

The principles behind MVD were first discussed by Louis and Rawlins [34], where multicriterion selection offers an implicit niching mechanism to facilitate diversity maintenance. Yet, the term "multiobjectivization" was coined by Knowles *et al.* [1], where the classic traveling salesman problem (TSP) is transformed into a bi-objective via MVD. Shi *et al.* [35] further studied the TSP, which was decomposed into two subproblems with a controllable correlation. They suggested that an appropriate correlation between subproblems provides more chances to escape from local optima.

To reduce the number of local optima, problem $f(\mathbf{x}) = f_1(\mathbf{x}) + f_2(\mathbf{x})$ was multiobjectivized into $\mathbf{F}(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x})]^T$ by Ishibuchi *et al.* [36] for the 0-1 knapsack problem. The number of local optima can be reduced by swapping the search space of different objectives in the multiobjectivization formulation [11]. Another example is a single-objective triangulation problem, which was multiobjectivized by Silva *et al.* [37] into a biobjective optimization problem. This multiobjectivization formulation finds better results than its SOP formulation, especially when only a few correspondence points in 3-D computer vision are available.

In addition to reducing the number of local optima, MVD can introduce plateaus of incomparable solutions studied by Handl *et al.* [38]. Problems can be made easier or harder depending on the extent of reducing such local optima in comparison to the extent of introducing plateaus. For example, a multiobjective hillclimber (MOHC) can perform better than a single-objective hillclimber (SOHC) on the short path function, but worse on the slope function. Furthermore, Handl *et al.* [39] analyzed how MVD changes the fitness landscape and affects the search. MOHC provides better guidance than SOHC toward the global optimum and away from the local optima on difficult protein structure prediction (PSP) problems. The set of solutions obtained by SOHC is usually a subset of those obtained by the hybrid hillclimber. They also compared the proposed decomposition with random decomposition, uninformative decomposition, and smooth decomposition.

To deal with the difficulty of specifying a value for parameter λ in $f(\mathbf{x}) = f_1(\mathbf{x}) + \lambda f_2(\mathbf{x})$, Gong *et al.* [40] multiobjectivized the learning problem of sparse features into a biobjective optimization problem $[f_1(\mathbf{x}), f_2(\mathbf{x})]^T$. Their results showed that the proposed multiobjective model can learn more useful sparse features. A similar multiobjectivization was proposed for compressed sensing in [41].

Different from the above simple and intuitive decompositions, Ceberio *et al.* [42] proposed elementary landscape decomposition for linear ordering, binary quadratic optimization, and frequency assignment problems. Each elementary landscape obtained from the decomposition is used as an independent objective to optimize. The experiments revealed that multiobjectivization based on elementary landscape decomposition outperforms single-objective methods.

A general MVD can be expressed by decomposing $f(f_1(\mathbf{x}), \ldots, f_m(\mathbf{x}))$ into $[f_1(\mathbf{x}), \ldots, f_m(\mathbf{x})]^T$. For example, Nebro *et al.* [43] decomposed the antenna placement problem $f(\mathbf{x}) = ([\text{Coverage}(\mathbf{x})]^\alpha/\text{number of antennae})$ into a biobjective optimization problem $[100 - \text{Coverage}(\mathbf{x}), \text{number of antennae}]^T$. Another example was presented by Churchill *et al.* [44], which decomposed the expensive tool selection problem $f(\mathbf{x}) = f_t(\mathbf{x}) + f_c(\mathbf{x})$ into a biobjective problem $[f_t(\mathbf{x}), f_d(\mathbf{x})]^T$, where $f_d(\mathbf{x})$ is a function of $f_c(\mathbf{x})$.

As a special MVD, MOS was first proposed by Jahne *et al.* [14], who divided the original objective into multiple subobjectives nested or with overlapping parts [14]. Specifically, they split the TSP into two even-sized sections randomly. Many random partitions can be created and the search can swap them dynamically. Three different metrics were proposed to create random partitions based on the expected value of distances (EVOD), the standard deviation of distances, or the expected value of different neighbors. Later, Lochtefeld and Ciarallo [13] proposed multiply spatial segmentations, such as radial segmentation, k-means segmentation, vertical segmentation, and hybrid segmentation. Their performance is better than Jahne's MOS method [14] and Knowles' decomposition method [1]. By experiments, they found that MOS methods with less multiobjectivization degree converge faster but can prematurely converge on TSPs [13]. Lochtefeld and Ciarallo [13] then proposed a more robust algorithm combining the different levels of segmentation and named it as multiobjectivization via progressive segmentation (MOPS). MOPS progressed from a more single-objective approach to a stronger multiobjectivization approach.

A simple sum of three components, that is: 1) code coverage; 2) crash coverage; and 3) stack trace similarity, is multiobjectivized into a triobjective optimization problem for generating reported crashes [45]. Results have indicated that for complex crashes, the proposed multiobjectivization method needs less test generation time than the SOO method.

There are also some attempts to multiobjectivize the problem into a many-objective optimization problem. An early example is Coello and Aguirre [46] decomposed the optimization objective of generating logical circuits into multiple subobjectives, each of which corresponds to an output of the circuits. A subpopulation is generated for each subobjective and these subpopulations cooperate to construct a solution that can produce correct outputs. Their experiments have demonstrated to produce better solutions than human designers and an SOO approach. In [10], the cost of a single-source shortest paths problem (SSSP) $\sum_{i=1}^{n-1} f_i(\mathbf{t})$ is multiobjectivized into $[f_1(\mathbf{t}), \ldots, f_{n-1}(\mathbf{t})]^T$, where $f_i(\mathbf{t})$ is the length from the root node n to the node i. This multiobjectivization method performs better on highly connected graphs, while the SOO method is solved faster on less connected graphs [47], [48].

B. Dynamic Decomposition Methods

Using static decomposition methods may bias the search. A solution to this issue is to use dynamic decomposition during the evolution [3], [13]. If there are multiply alternative decompositions and have no knowledge about which decomposition is best, it is reasonable to switch decompositions after certain generations so that the multiobjectivization does not overly bias the search [7], [8], [13].

For example, Garza-Fabre *et al.* [49] proposed a random decomposition for the PSP problem. Each H amino acid is randomly associated with different objectives of the MVD problem. They showed that this random decomposition method outperforms the SOO method and a static decomposition method. After that, they [8] further decomposed the energy (objective) function of the PSP problem into two subcomponents considered as two objectives of the MVD problem based on three decomposition schemes, that is: 1) the parity

TABLE II
SUMMARY OF STUDIES ON MVD METHODS, WHERE *n* IS THE NUMBER OF DECISION VARIABLES

Work	Original problem	MVD strategies	Advantages	Disadvantages
Static decomposition into a bi-objective optimization problem				
A deceptive problem [34]	$f(\mathbf{x}) = f_1(\mathbf{x}) + f_2(\mathbf{x})$	$[f_1(\mathbf{x}), f_2(\mathbf{x})]^T$	Escape from local optima and increase paths to the global optima	Static decomposition
TSP [1]	$f(\pi) = \sum_{i=1}^{n} d_{c_{\pi[i]}, c_{\pi[i\oplus 1]}}$ $= f_1(\pi, a, b) + f_2(\pi, a, b)$	$[f_1(\pi, a, b), f_2(\pi, a, b)]^T$	The term "multi-objectivization" is first proposed here	The choice of city pairs influences the performance [13, 14]
KP [36], UBQP [35], Triangulation[37]	$f(\mathbf{x}) = f_1(\mathbf{x}) + f_2(\mathbf{x})$	$[f_1(\mathbf{x}), f_2(\mathbf{x})]^T$	NSGA-II outperforms SOGA in most problems [36]	Should be validated in more problems [35, 37]
Protein structure prediction [39]	$f(\mathbf{x}) = f_1(\mathbf{x}) + f_2(\mathbf{x})$	$[f_1(\mathbf{x}), f_2(\mathbf{x})]^T$, or $[f_1(\mathbf{x}) + f_2(\mathbf{x}) - r(\mathbf{x}), r(\mathbf{x})]^T$	Random, smooth, and uninformative decomposition are proposed	Static decompositions
Slope function [38]	$f(\mathbf{s}) = f_1(\mathbf{s}) + f_2(\mathbf{s})$	$[f_1(\mathbf{s}), f_2(\mathbf{s})]^T$, $f_1(\mathbf{s}) = \lvert\mathbf{s}\rvert_1$, $f_2(\mathbf{s}) = 2n - \lvert\mathbf{s}\rvert_1$	The expected runtime of multi-objectivization is $\Theta(2^n)$ [38]	The expected runtime of SOO is $\Theta(n \log n)$ only [38]
Short path problem [38]	$f(\mathbf{s}) = f_1(\mathbf{s}) + f_2(\mathbf{s})$	$[f_1(\mathbf{s}), f_2(\mathbf{s})]^T$	The expected runtime of multi-objectivization is $\Theta(n^3)$ [38]	The expected runtime of SOO is $n^{\Omega(n)}$ [38]
Sparse learning [40, 41]	$f(\mathbf{x}) = f_1(\mathbf{x}) + \lambda f_2(\mathbf{x})$	$[f_1(\mathbf{x}), f_2(\mathbf{x})]^T$	This multi-objectivization is more useful	Should furthermore be validated in large-scale problems
Antenna placement [43]	$f(\mathbf{x}) = \frac{[Coverage(\mathbf{x})]^\alpha}{\lvert Antennaes\rvert}$	$[100 - Coverage(\mathbf{x}), \lvert Antennaes\rvert]^T$	Allowing the DM to choose the best coverage/cost tradeoff solution	Its effectiveness is studied on radio network design only [10]
Tool selection problem [44]	$f(\mathbf{x}) = f_t(\mathbf{x}) + f_c(\mathbf{x})$,	$[f_t(\mathbf{x}), f_d(\mathbf{x})]^T$, $f_d(\mathbf{x})$ is a function of $f_c(\mathbf{x})$	Find many better solutions than the SOO algorithm	Static decomposition
LOP [42]	$f(\sigma) = \sum_{i,j} b_{\sigma(i),\sigma(j)}$	Elementary decomposition $[f_1(\sigma), f_2(\sigma)]^T$, $f_1(\sigma) = \sum_{i\neq j, p\neq q} b_{p,q}\Omega^1_{i,j,p,q}(\sigma)/(2n)$, $f_2(\sigma) = \sum_{i\neq j, p\neq q} b_{p,q}\Omega^3_{i,j,p,q}(\sigma)/(n^2 - 2n)$	This decomposition provides a new framework to multi-objectivize	The neighbourhood system needs certain symmetry and regularity properties [42]
UQO [42]	$f(\mathbf{x}) = \sum_{i,j} q_{i,j} x_i x_j$	Elementary decomposition $[f_1(\mathbf{x}), f_2(\mathbf{x})]^T$, $f_1(\mathbf{x}) = -\sum_{i,j}(q_{ij} + q_{ji})(1 - 2x_i)/4$, $f_2(\mathbf{x}) = \sum_{i,j} q_{ij}(1 - 2\lvert x_i - x_j\rvert)/4$	This decomposition provides a new framework to multi-objectivize	The neighbourhood system needs certain symmetry and regularity properties [42]
FAP [42]	$f(\mathbf{x}) = \sum_{i,j} w_{i,j}^{x_i, x_j}$	Elementary decomposition $[f_1(\mathbf{x}), f_2(\mathbf{x})]^T$, $f_1(\mathbf{x}) = \sum_{i\neq j}^{n} \sum_{p,q=1}^{r} w_{i,j}^{p,q} \phi_{i,j,r-2}^{p,q}(\mathbf{x})/r$, $f_2(\mathbf{x}) = -\sum_{i\neq j}^{n} \sum_{p,q=1}^{r} w_{i,j}^{p,q} \phi_{i,j,-2}^{p,q}(\mathbf{x})/r$ $+ \sum_{i=1}^{n} \sum_{p=1}^{r} w_{i,i}^{p,p} \delta_{x_i}^{p}$	This decomposition provides a new framework to multi-objectivize	The neighbourhood system needs certain symmetry and regularity properties [42]
TSP [14]	$\sum_{i=1}^{n} d(c_{\pi[i]}, c_{\pi[i\oplus 1]})$	EVOD, SDD, and EVDN segmentations	MOS strategies outperform SOO and Knowles' approach [1]	Random adaptive segmentation is used [13]
TSP [13]	$\sum_{i=1}^{n} d(c_{\pi[i]}, c_{\pi[i\oplus 1]})$	Radial, k-means segmentation, NNROD, vertical, and hybrid segmentation	Introduction of the SNR rule to select the segment	Tested on TSPs only
Static decomposition into a tri-objective optimization problem				
Reported crashes [45]	$f(\mathbf{t}) = d_s(\mathbf{t}) + d_{except}(\mathbf{t})$ $+ d_{trace}(\mathbf{t})$	$[d_s(\mathbf{t}), d_{except}(\mathbf{t}), d_{trace}(\mathbf{t})]^T$	Multi-objectivization needs less test generation time than SOO	Should furthermore be validated [45] in other applications
QAP [42]	$f(\sigma) = \sum_{ij} d_{ij} h_{\sigma(i),\sigma(j)}$	$f_1(\sigma) = \sum_{i\neq j, p\neq q} d_{i,j} h_{p,q} \Omega^1_{ijpq}(\sigma)/(2n)$, $f_2(\sigma) = \sum_{i\neq j, p\neq q} d_{i,j} h_{p,q} \Omega^2_{ijpq}(\sigma)/(2n - 4)$, $f_3(\sigma) = \sum_{i\neq j, p\neq q} d_{i,j} h_{p,q} \Omega^3_{ijpq}(\sigma)/(n^2 - 2n)$	This decomposition provides a new framework to multi-objectivize	The neighbourhood system needs certain symmetry and regularity properties [42]
Static decomposition into a many-objective optimization problem				
Circuit design [46]	$f(\mathbf{x})$	Each output of the circuits is an objective	It outperforms human designers and a SOO approach	Too many objectives may be introduced
SSSP [10, 47, 48]	$f(\mathbf{t}) = \sum_{i=1}^{n-1} f_i(\mathbf{t})$	$[f_1(\mathbf{t}), \ldots, f_{n-1}(\mathbf{t})]^T$	Multi-objectivization performs better on highly connected graphs	SOO is faster on less connected graphs [10]
Dynamic decomposition				
JSSP [7]	$f_\Sigma(\mathbf{x}) = \sum_{i=1}^{n} f_i(\mathbf{x})$	$[f_1(\mathbf{x}), \ldots, f_n(\mathbf{x})]^T$	Complete decomposition	Too many objectives are optimized simultaneously [13]
Protein structure prediction[49]	$f(\mathbf{x}) = \sum_{a_i, a_j} e(a_i, a_j)$	Parity decomposition $[f_1(\mathbf{x}), f_2(\mathbf{x})]^T$, $f_1(\mathbf{x}) = \sum_{a_i, a_j} e(a_i, a_j), i \equiv 0 (mod\ 2)$ $f_2(\mathbf{x}) = \sum_{a_i, a_j} e(a_i, a_j), i \equiv 1 (mod\ 2)$	Decomposition is based on a heuristic rule	Consider parity only [8]
Protein structure prediction[8]	$f(\mathbf{x}) = \sum_{a_i, a_j} e(a_i, a_j)$	Locality decomposition $[f_1(\mathbf{x}), f_2(\mathbf{x})]^T$, $f_1(\mathbf{x}) = \sum_{a_i, a_j} e(a_i, a_j), i < j \leq i + \delta$ $f_2(\mathbf{x}) = \sum_{a_i, a_j} e(a_i, a_j), i + \delta < j$	Decomposition is based on a heuristic rule	Consider locality only [8]
Protein structure prediction[50]	$f(\mathbf{x}) = \sum_{a_i, a_j} e(a_i, a_j)$	H-subsets decomposition $[f_1(\mathbf{x}), f_2(\mathbf{x})]^T$, $f_1(\mathbf{x}) = \sum_{k=1}^{2} \sum_{a_i, a_j \in H_k} e(a_i, a_j)$ $f_2(\mathbf{x}) = \sum_{a_i \in H_1, a_j \in H_2} e(a_i, a_j)$	Decomposition is based on a heuristic rule	Consider H-subsets only [8]

decomposition [49]; 2) the locality decomposition (LD) [8]; and 3) the H-subsets decomposition (HD) [50]. These decomposition methods have been shown to outperform the SOO method. The HD method performed the best on 2-D cases, while the LD method scored the best results on 3-D cases.

There are several key parameters used to define and switch different decompositions, such as the number of decompositions, the number of generations to switch decomposition, which and how many subcomponents in each decomposed subproblems [13]. Lochtefeld and Ciarallo [13] investigated the effect of different degrees of decomposition (the number of objectives in the MVD problem) on the algorithmic performance. They have shown that a progressive multiobjectivization from a more SOO method to a strong multiobjectivization method can obtain a more robust performance. Different numbers of generations to switch decomposition are explored in [49]. A summary of decomposition methods is presented in Table II.

IV. MULTIOBJECTIVIZATION VIA HELPER-OBJECTIVES

MHO methods add helper-objectives together with the original problem to form the MOP. The original problem and the additive helper-objectives are solved simultaneously with the expectation of achieving a better result than merely focusing on the original problem [16].

This section introduces MHO methods from three aspects, that is: 1) static helper-objectives; 2) helper-objectives based on random selection; and 3) helper-objectives based on adaptive selection.

A. Static Helper-Objectives

This work for the first time categorizes the helper objective used in MHO methods into four groups, that is: 1) a variation of the original objective $f(\mathbf{x})$; 2) a subobjective of $f(\mathbf{x})$; 3) $f(\mathbf{x})$ with a different value of the key parameter; and 4) an auxiliary objective without a direct relationship with $f(\mathbf{x})$.

1) Helper-Objective Is Variation of the Original Objective: It is a natural idea to construct the help-objective based on the original objective. The first work in this direction was proposed by Abbass and Deb [51], which studied multiobjectivization via adding $1/f(\mathbf{x})$ as the helper-objective. It has shown that this helper-objective can improve the population diversity and reduce the selection pressure toward the main objective [51]. Besides, without the mutation operator, the multicriteria approach can also outperform the single-criterion approach. In other words, the multiobjectivization method is less sensitive to the choice of the mutation probability than the SOO method due to its natural diversity control.

Later, Watanabe and Sakakibara [52] tried to define the helper-objective by adding a noise to the objective value $h(\mathbf{x}) = f(\mathbf{x}) + \varepsilon$ or decision variables $h(\mathbf{x}) = f(\mathbf{x} + \varepsilon)$, where ε is a small real number and $f(\mathbf{x})$ is the original objective. These helper-objectives can enhance the population diversity and introducing one of the helper-objectives can achieve a better result than considering the original objective $f(\mathbf{x})$ alone.

The helper-objective $\mathrm{OneMax}(\mathbf{x})$ is used to optimize the strongly correlated objective $\lfloor \mathrm{OneMax}(\mathbf{x})/k \rfloor$ by Buzdalov and Buzdalova [53], where k is a divisor of the number of decision variables. Their results have indicated that using the strongly correlated objective as the helper can assist the solving of the original objective.

Thomas and Jin [54] added an equivalent frequency-domain-based helper-objective to help optimize the original objective, which is a time-domain function. They found that adding an equivalent helper-objective in the frequency domain improves the solving performance, compared to optimizing the original objective solely.

Theoretical analyses are also proposed for MHOs. Researchers showed that introducing the helper-objective(s) may make a problem harder or easier. On the SetCover problem, Friedrich *et al.* [55] proved that optimizing $[u(\mathbf{x}), u(\mathbf{x}) + w(\mathbf{x})]^T$ with an exponential runtime is worse than optimizing the main objective $u(\mathbf{x})$ solely with a polynomial-time, where $w(\mathbf{x}) = \sum_{i=1}^{n} c_i x_i$ denotes the total cost of the selection and $u(\mathbf{x})$ is the number of uncovered elements. However, for a more general SetCover problem, they [56] verified that optimal solutions can be approximated within a logarithmic time by using the MHO model but a polynomial time by using the SOO model.

Aiming to locate all global and local optima in multimodal optimization, Yao *et al.* [57] introduced $([\|\nabla f(\mathbf{x})\|_1]/n)$ as the helper-objective optimized with the main objective $f(\mathbf{x})$ simultaneously, since $\|\nabla f(\mathbf{x})\|_1 = 0$ is the essential property of all inner local optima. The experimental studies have revealed that introducing the gradient information as the helper-objective performs better than other niching techniques in locating all global and local optima.

To enhance the exploration ability, an augmented function $f(\mathbf{x}) + \lambda \sum_{i \in F} p_i I_i(\mathbf{x})$ is added as the helper-objective to help optimize the main objective $f(\mathbf{x})$ in [58], where F is the set of all features, p_i is a penalty factor of the ith feature, and $I_i(\mathbf{x})$ is an indicative function. In this way, the algorithm searches for the best tradeoff between the main objective and its augmented objective. This multiobjectivization can keep good exploitation for the main objective and certain exploration due to the addition of the helper-objective.

2) Helper-Objective Is Similar to the Original Objective But With Different Parameter/Factor: Key parameters/factors play an important role in the optimization objective, especially when this objective is approximate rather than true. To relieve the effect of the key parameter/factor on the original objective, helper-objectives can be constructed based on different alternative parameter values/factors to assist the search and robustness. For example, Brys *et al.* [9] multiobjectivized the reinforcement learning by injecting different reward shaping functions to create enriched reward signals, which can be combined strategically based on ensemble techniques to reduce sample complexity.

In subset selection based on the hypervolume indicator, the selected result is strongly dependent on a single reference point. To deal with this issue, Ishibuchi *et al.* [59] multiobjectivased this problem as an MOP by using multiple reference points to construct helper-objectives for the subset selection.

A single cooperator/competitor is usually used to estimate the fitness of a subcomponent/individual in coevolutionary algorithms. However, the performance of this fitness evaluation is strongly dependent on cooperators/competitors. To address this issue, multiple helper-cooperators/competitors, instead of a single one, are used and individual comparison is then based on Pareto-dominance. Bucci and Pollack [60] found that the multiple-cooperators algorithm obtains better than the single-cooperator algorithm. Noble and Watson [61] showed that the multiple-competitors algorithm outperforms a single-competitor algorithm on game theory.

3) Helper-Objective Is Subobjective of the Original Objective: The idea of using subobjective/subcost as the helper-objective is inspired by that it may be useful to reward diversification steps containing at least one subobjective improved [62]. A current move in a negative direction for the main objective may have the potential to help a future longer improvement [62]. For example, Wright [62] changed the probability of accepting a worse solution based on both the improvement in subcosts and the degeneration in the original objective in the simulated annealing.

Working with a well-known plateau function, Brockhoff *et al.* [4], [63] illustrated that adding helper-objectives can make a problem harder or easier. Using the subobjective $\|\mathbf{x}\|_0$ as the help-objective can help optimize the Plateau function. On the contrary, adding an irrelevant help-objective $\|\mathbf{x}\|_1$ can make the Plateau function harder.

Lochtefeld and Ciarallo [2] used the subobjective as the helper-objective to study the signal-to-noise (SNR) principle, which stemmed from some fitness improvements in a subobjective/helper-objective $f_1(\mathbf{x})$ (signal) to be not always recognized and rewarded if they happen in a solution with a fitness decrement (noise) in other parts $f(\mathbf{x}) - f_1(\mathbf{x})$ of the objective $f(\mathbf{x})$. SNR is determined by the ratio of fitness improvements on $f_1(\mathbf{x})$ to the fitness decrements on $f(\mathbf{x}) - f_1(\mathbf{x})$. It was further studied by Lochtefeld and Ciarallo [2] to find fundamental drivers of the MHO.

4) Helper-Objective Is Auxiliary Objective: Improving the generalization ability is widely used as the 2nd objective in the fields of machine learning. For example, Kim [64] investigated the effect of using the tree size as the 2nd objective in designing the decision tree. This multiobjectivization method aims at finding the minimum misclassification error for each tree size and outperforms a classical single-objective optimizer. To balance the number and the contained formation of the selected bands, Gong *et al.* [67] multiobjectivized the band selection problem into a biobjective optimization problem. In genetic programming, its goal is to evolve an accurate program tree to solve specific tasks. The helper-objective is the minimization of the tree size [65], [66]. This multiobjectivization maintains trees of different sizes to promote diversity and allows small trees with suitable functionality.

Minimizing the model complexity is often used as the 2nd objective in the fields of optimization. Greiner *et al.* [68] introduced the number of cross section types as the helper-objective to optimize the structure design problem. By adding helper-objectives, less emphasis may be put on tuning the mutation rate [11], [20]. In [69], the vehicle routing problem (VRP) is optimized together with the number of routes as the helper-objective. This multiobjectivization method obtains better and more robust solutions. Using the minimum spanning tree, Neumann and Wegener [10], [70] proved that the multiobjectivization formulation $[c(\mathbf{s}), w(\mathbf{s})]^T$ with the expected time $O(en^2)$ performs better on highly connected graphs, while the original objective $w(\mathbf{s})$ with the expected time $O(e^2 \log n)$ is faster on less connected graphs. Here, $w(\mathbf{s})$ is the total weight of all selected edges, $c(\mathbf{s})$ is the number of connected parts of the graph described by $\mathbf{s}$, n, and e are, respectively, the number of nodes and edges.

Domain knowledge is often used to construct the helper-objective. It is usually strongly related to the original problem and implies direct or indirect features on the optimal solution to the original problem. For example, Sharma *et al.* [71] introduced a preference-based helper-objective in the compliant mechanism design. The main objective is the weight of the structure and the helper-objective is the geometrical dissimilarity of structure to the reference design. This helper-objective is selected in conflict with the main objective, thereby maintaining the population diversity. Another example is the short-term unit commitment problem, which is multiobjectivized by using the reliability as the helper-objectives in [72]. Two used helper-objectives are the expected unserved energy (EUE) and the loss of load probability (LOLP). This multiobjectivized method has shown to find many better solutions (in terms of the original objective) than the SOO method. For the autotuning of programs, the energy consumption was used by Durillo and Fahringer [73] as the helper-objective to help optimize the main objective, that is, the execution time. A summary of multiobjectivization via static helper-objective methods is presented in Table III.

B. Helper-Objectives Based on Random Selection

Using the static helper-objectives may bias the search. Besides, the best helper-objective for a given problem is usually unknown in advance. A solution to this issue is to use dynamic helper-objectives [3], [16], which benefit the search since changes of the search space can facilitate escaping from local optima [15], [21]. The used sequence of helper-objectives has an important influence on the search [3]. Thus, a few studies are proposed to use helper-objectives based on a random selection.

Jensen [3], [16] made the first try to change helper-objectives dynamically during the evolution. Each helper-objective sums the cost of incoming and outgoing links on its associated cities in solving the TSP. For the job shop scheduling problem (JSSP), they suggested using the subobjective as the helper-objective when this original objective is of a sum or product of many subobjectives. Due to a lack of knowledge about the optimal helper-objective sequence, a subset of helper-objectives is randomly chosen and used simultaneously for each period [3], [16]. Adding a small number of dynamic helper-objectives simultaneously is promising, since adding too many helper-objectives makes it become a many-objective optimization problem reducing the selection pressure toward the original problem.

Another random helper-objective selection was proposed by Buzdalova and Buzdalov [74], who checked whether a stagnation occurred for a certain amount of iterations. If so, a random helper-objective is used. Otherwise, the previously selected helper-objective is used. Another way is that the main objective is optimized together with a randomly selected helper-objective no matter whether a stagnation occurs. A summary of MHOs based on a random selection is given in Table IV.

C. Helper-Objectives Based on Adaptive Selection

No feedback information used during the evolution is one of the main defects in static helper-objectives and most random helper-objectives. The careful selection and the smart switching of helper-objecitves are potential directions [11]. Recently, the adaptive helper-objective selection based on a heuristic strategy has attracted increasing attention [20].

Lochtefeld and Ciarallo [11] made the first try to explore the effect of different helper-objective sequences. A good helper-objective sequence can be obtained by domain knowledge, for example, the shortest job first (SJF) in [11] and [20]. The SJF strategy sequences helper-objectives such that the helper-objective associated with the shortest job is first selected, the helper-objective associated with the next shortest job is second selected, and so forth. This SJF strategy outperformed the random helper-objective selection strategy and was further studied in [7].

TABLE III
SUMMARY OF STUDIES ON MULTIOBJECTIVIZATION VIA STATIC HELPER-OBJECTIVE METHODS

Work	Original problem	Static helper-objectives	Advantages	Disadvantages
The helper-objective is a variation of the original objective				
Griewangk and Rastrigin [51]	$f(\mathbf{x})$	$\frac{1}{f(\mathbf{x})}$	With no mutation, multi-objectivization outperforms the SOO	Strong diversity [51]
Schwefel, Ridge, and Rastrigin [52]	$f(\mathbf{x})$	$f(\mathbf{x}+\varepsilon)$ or $f(\mathbf{x})+\varepsilon$	This multi-objectivization achieves a better result than the SOO	Strong randomness [52]
XdivK problem [53]	$\lfloor \text{OneMax}(\mathbf{x})/\text{k} \rfloor$	$\text{OneMax}(\mathbf{x})$	The helper may assist the optimization of its strongly related function	Should be validated on more problems
Gene regulatory networks [54]	f_{t_s}	f_{w_s} or f_{w_u}	Helper-objective in frequency domain help the time-domain objective	Adding a time-domain objective may not help the frequency-domain objective [54]
Multimodal optimization [57]	$f(\mathbf{x})$	$\frac{\Vert\nabla f(\mathbf{x})\Vert_1}{n}$	This multi-objectivization is better than other SOGAs in locating all global and local optima	Not work for non-differentiable or discrete functions [57]
SetCover problem [55]	$u(\mathbf{x})$	$u(\mathbf{x})+w(\mathbf{x})$	Using $u(\mathbf{x})+w(\mathbf{x})$ instead of $w(\mathbf{x})$ as the helper can reduce incomparable solutions [55]	The runtime of the multi-objectivization is worse than the SOO [55]
TSP and QAP [58]	$f(\mathbf{x})$	$f(\mathbf{x})+\lambda\sum_{i\in F}p_i I_i(\mathbf{x})$	This multi-objectivization outperforms SOO and multiple multi-objectivization methods	The search have a strong exploitation of the main objective [58]
Plateau1 function [55]	$Plateau1(x)$	$PL(\mathbf{x})$	Show the possibility that multi-objectivization can make a problem harder	Exponential runtime for MOP but polynomial runtime for SOP [55]
The helper-objective is similar to the original objective but with a different parameter/factor				
Reinforcement learning [9]	$E\big[\sum\limits_{t=0}^{\infty}\gamma^t R(s_t,a_t,s_{t+1})\big]$	$\Big[E\big[\sum\limits_{t=0}^{\infty}\gamma^t R_1(s_t,a_t,s_{t+1})\big],\ldots, E\big[\sum\limits_{t=0}^{\infty}\gamma^t R_m(s_t,a_t,s_{t+1})\big]\Big]^T$	Multi-objectivization helps ensemble techniques to reduce the sample complexity	Hard to build a shaping generator similar to a weak learner [9]
Set optimization [59]	Subset selection based on a reference point	Different reference points are used as different helper-objectives	The results are not sensitive to the choice of reference point	Need to specify a set of reference points [59]
MTQ [60]	Based on a single-cooperator (the best individual)	Based on more helper-cooperators selected from the population	Find the global optima more reliably	Test problems are too simple
Game [61]	Strategy compete with a strategy	The strategies compete against each other	Pareto-based selection outperforms single fitness selection	Pareto-based selection may not hold onto their collective wisdom over time [61]
The helper-objective is a sub-objective of the original objective				
Timetabling problem [62]	$f(\mathbf{x})=\sum_{i=1}^{4}f_i(\mathbf{x})$	Subcosts $f_1(\mathbf{x}),\ldots,f_4(\mathbf{x})$	Increasing the accepting probability of an inferior solution but good on one subcost	Fixed subcosts
PLATEAU1 [4, 63]	$f(\mathbf{x})$	$\Vert\mathbf{x}\Vert_0$ or $\Vert\mathbf{x}\Vert_1$	Different helpers make a problem harder or easier depending on the problem structure	No clear suggestion on how to select the helper-objective
TOP [2]	The Weise model	Subordinate-objectives and subordinate objective groups	Show the fundamental drivers of the multi-objectivization via helpers	This abstract problem is hard to understand
The helper-objective is an auxiliary objective				
Decision tree optimization [64]	Misclassification error	The tree size	Find the minimum misclassification error for each size of decision trees	Take more computing time than SOO [64]
Genetic [65] programing [66]	$f(\mathbf{x})$	The program size	Reducing the program size and promoting diversity	Should furthermore be validated in problems with larger scale
Image process [67]	Classification accuracy	$[length(\mathbf{x}), 1/[\sum_{i=1}^{length(\mathbf{x})}h(\mathbf{x}_i)]]^T$	Obtain more diverse band subsets and better performance on classification	Should furthermore be validated in large-scale problems [67]
Computational mechanics [68]	The constrained mass	The number of different cross-section types	The helper-objective method may be less sensitive to the mutation rate	Static helper-objective based on domain knowledge [68]
VRP [69]	The total travel distance	A helper-objective on the assignment of customers	Multi-objectivization can find the solutions with better quality and less variation	Static helper–objective method
MST [10, 70]	$c(\mathbf{s})$	$w(\mathbf{s})$	Multi-objectivization performs better on highly connected graphs	SOO is faster on less connected graphs [10, 70]
Structural topology optimization [71]	Weight of structure	Geometrical dissimilarity of structure on the reference design	The helper is conflict with the main objective, thereby maintaining the population diversity	The helper-objective is fixed
Power system planning [72]	A economic objective	Two helpers are LOLP and EUE reliability indices	Find many better solutions than the SOO algorithm	Static helper-objectives
Auto-tuning of programs [73]	Execution time	Energy consumption	Multi-objectivization can obtain a better result than SOO	The helper-objective is fixed
VertexCover problem [56]	$u(\mathbf{x})$	$\Vert\mathbf{x}\Vert_1$	Polynomial runtime for multi-objectivization but exponential runtime for SOO	A limited number of applications are tested
General SetCover problem [56]	$u(\mathbf{x})$	$p(\mathbf{x})$	Logarithmic runtime for multi-objectivization but exponential runtime for SOO	A limited number of applications are tested
SetCover problem [55]	$u(\mathbf{x})$	$p(\mathbf{x})$	Show the possibility of multi-objectivization making a problem harder	SOO is better than multi-objectivization [55]

TABLE IV
SUMMARY OF STUDIES ON MHOS BASED ON A RANDOM SELECTION

Work	Original problem	Helper-objectives	Advantages	Disadvantages
JSSP [3, 16]	$f_\Sigma(\mathbf{x})=\sum_{i=1}^{n}f_i(\mathbf{x})$	A subset of helper-objectives is randomly chose from $\{f_1(\mathbf{x}),\ldots,f_n(\mathbf{x})\}$	Introduce dynamic helper-objectives during the search	Random helper-objectives are used
TSP [3]	$\sum_{i=1}^{n}d(c_{\pi[i]},c_{\pi[i\oplus 1]})$	$h(\pi,\mathbf{p})=\sum_{i=1}^{\vert\mathbf{p}\vert}d(c_{\pi[\pi^{-1}[\mathbf{p}[i]]\ominus 1]},c_{\pi[i]}) + d(c_{\mathbf{pi}[i]},c_{\pi[\pi^{-1}[\mathbf{p}[i]]\oplus 1]})$, p is a subset of $\{1,\ldots,n\}$ seleced randomly	Introduce dynamic helper-objectives during the search	Random helper-objective is used
H-IFF [75]	$f(\mathbf{x})=f_0(\mathbf{x})+f_1(\mathbf{x})$	Randomly choose a helper-objective from $\{f_0(\mathbf{x}),f_1(\mathbf{x})\}$	Introduce dynamic helper-objectives during the search	Strong randomness without problem characteristic used [76]

Reinforcement learning (RL) is another alternative for adaptive helper-objective selection [74], [76], which automatically selects the most effective helper-objective and ignores ineffective ones. Different from the problem-dependent SJF strategy [7], [20], the RL strategy is usually independent of the problem and hence increases the generality of the helper-objective method. As the pioneers, Buzdalova and Buzdalov [74] used the RL to select the optimization objectives from the helper-objectives and the main objective for evolution. The selected

TABLE V
SUMMARY OF STUDIES ON MULTIOBJECTIVIZATION METHODS VIA ADAPTIVE HELPER-OBJECTIVES

Work	Original problem	Helper-objectives	Advantages	Disadvantages
JSSP [11, 20]	$f_\Sigma(\mathbf{x}) = \sum_{i=1}^n f_i(\mathbf{x})$	$f_1(\mathbf{x}), \ldots, f_n(\mathbf{x})$	A small number of helper-objectives based on the SJF strategy obtains the best performance	The SJF strategy is dependent on the problem [74]
JSSP [7]	$f_\Sigma(\mathbf{x}) = \sum_{i=1}^n f_i(\mathbf{x})$	$f_1(\mathbf{x}), \ldots, f_n(\mathbf{x})$	The SJF helper-objective selection outperforms random helper-objective selection	The SJF strategy is dependent on the problem [74]
OneMax problem [74]	$OneMax(\mathbf{x})$	$\lfloor OneMax(\mathbf{x})/k \rfloor$,$ZeroMax(\mathbf{x})$	RL is used to automatically select the most effective helper-objective	Should furthermore be validated in more problems
XdivK problem [53]	$\lfloor OneMax(\mathbf{x})/k \rfloor$	$OneMax(\mathbf{x})$	The helper-objective may facilitate optimizing its strongly related function	Should be validated in more problems
Scheduling problem [18]	The tied-up capital	The number of parts produced in the 1st step, and the total buffer capacity in the 2nd step	This multi-objectivization outperforms an SOO method and a static helper method	Two helper-objectives are fixed and may bais the search

TABLE VI
SUMMARY OF STUDIES ON MSFS, WHERE $\mathbf{w} = (w_1, \ldots, w_m)$ IS A WEIGHT VECTOR

Work	Original problem	MSF strategies	Advantages	Disadvantages
KP [36]	$\mathbf{F}(\mathbf{x}) = [f_1(\mathbf{x}), \ldots, f_m(\mathbf{x})]^T$	Multiple runs of SOGA to solve different scalar functions $\sum_{i=1}^m w_i f_i(\mathbf{x})$	Better diversity on bi-objective KPs and better convergence on many-objective KPs	Need to predefine weight vector $\mathbf{w}$ appropriately
KP [78]	$\mathbf{F}(\mathbf{x}) = [f_1(\mathbf{x}), \ldots, f_m(\mathbf{x})]^T$	Introduce a set of evenly scalarizing functions	Probabilistic use of scalarizing functions for parent selection and generation update	Need to predefine weight vectors appropriately [78]
KP [6]	$f_1(\mathbf{x}) + f_2(\mathbf{x})$	$[(w_1+\beta)f_1(\mathbf{x}) + (w_2-\beta)f_2(\mathbf{x}),$ $(w_1-\beta)f_1(\mathbf{x}) + (w_2+\beta)f_2(\mathbf{x})]^T$	Preferred search region of the construct MOP is near the optimum of the original problem	Introduce parameters $\mathbf{w}$ and β
KP and fuzzy system design [19]	$[f(\mathbf{x}), g(\mathbf{x})]^T$	$[f(\mathbf{x}) - w \cdot g(\mathbf{x}), f(\mathbf{x}) + w \cdot g(\mathbf{x}),$ $g(\mathbf{x}) - w \cdot f(\mathbf{x}), g(\mathbf{x}) + w \cdot f(\mathbf{x})]^T$	The value increase of w leads to the increase in the diversity of solutions	Convergence is degraded on four-objective problems

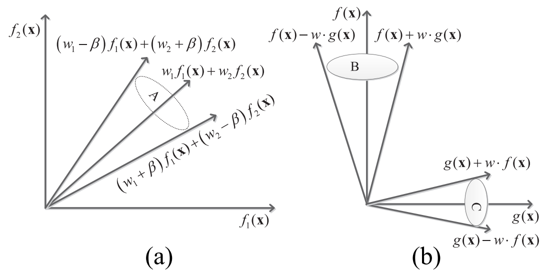

$f_2(\mathbf{x})$
$(w_1-\beta)f_1(\mathbf{x})+(w_2+\beta)f_2(\mathbf{x})$
$w_1f_1(\mathbf{x})+w_2f_2(\mathbf{x})$
A
$(w_1+\beta)f_1(\mathbf{x})+(w_2-\beta)f_2(\mathbf{x})$
$f_1(\mathbf{x})$
(a)
$f(\mathbf{x})$
$f(\mathbf{x})-w\cdot g(\mathbf{x})$
$f(\mathbf{x})+w\cdot g(\mathbf{x})$
B
$g(\mathbf{x})+w\cdot f(\mathbf{x})$
C
$g(\mathbf{x})$
$g(\mathbf{x})-w\cdot f(\mathbf{x})$
(b)

Fig. 3. (a) New generated objectives $(w_1+\beta)f_1(\mathbf{x}) + (w_2-\beta)f_2(\mathbf{x})$ and $(w_1-\beta)f_1(\mathbf{x})+(w_2+\beta)f_2(\mathbf{x})$, where w_1, w_2, and β are scalar parameters and $w_1+w_2 = 1$. (b) Relation between the original two objectives $f(\mathbf{x})$, $g(\mathbf{x})$ and the generated four objectives $f(\mathbf{x}) - w \cdot g(\mathbf{x})$, $f(\mathbf{x}) + w \cdot g(\mathbf{x})$, $g(\mathbf{x}) - w \cdot f(\mathbf{x})$, and $g(\mathbf{x}) + w \cdot f(\mathbf{x})$, where w is a scalar parameter.

helper-objectives are rewarded based on their fitness improvement.

A two-step multiobjectivization method [18] is proposed for scheduling problems. The first step adds a helper-objective in conflict with the main objective to promote exploration. The second step introduces another helper-objective in agreement with, but subservient to, the main objective to promote exploitation. Results have shown that this two-step multiobjectivization method outperforms the SOO method and a fixed helper-objective method. A summary of multiobjectivization methods via adaptive helper-objectives is presented in Table V.

V. MULTIOBJECTIVIZATION VIA SCALARIZING FUNCTIONS

MSFs is initially introduced for $f(\mathbf{x}) = f_1(\mathbf{x}) + \cdots + f_m(\mathbf{x})$ and $\mathbf{F}(\mathbf{x}) = [f_1(\mathbf{x}), \ldots, f_m(\mathbf{x})]^T$, which can be transformed into an MOP by multiple scalarizing functions $[\sum_{i=1}^m w_i f_i(\mathbf{x}), \ldots, \sum_{i=1}^m u_i f_i(\mathbf{x})]^T$, where $\mathbf{w} = (w_1, \ldots, w_m), \ldots, \mathbf{u} = (u_1, \ldots, u_m)$ are different weight vectors. Multiple prespecified weight vectors in MSF can serve as the preferences of the DM that enable the construction of a well-designed MOP by specifying weight vectors [11], [18].

Multiobjectivization can also assist the optimization of a scalarizing function. To avoid getting stuck in local optima in solving $w_1f_1(\mathbf{x}) + w_2f_2(\mathbf{x})$, Ishibuchi and Nojima [6] multiobjectivized it as a new MOP

$$\begin{bmatrix}(w_1+\beta)f_1(\mathbf{x}) + (w_2-\beta)f_2(\mathbf{x})\\(w_1-\beta)f_1(\mathbf{x}) + (w_2+\beta)f_2(\mathbf{x})\end{bmatrix} \tag{2}$$

by introducing a scalar parameter β as shown in Fig. 3(a). In the MOP (2), multiple different but similar objectives to $\sum_{i=1}^m w_i f_i(\mathbf{x})$ are constructed and optimized, so that the preferred search region A of the construct MOP (2) is near the optimal solution to $\sum_{i=1}^m w_i f_i(\mathbf{x})$ as shown in Fig. 3(a). Compared with the given scalarizing function $\sum_{i=1}^2 w_i f_i(\mathbf{x})$, two similar objectives are optimized simultaneously can remedy the issue of slow convergence on NP-hard knapsack problems.

Later, MSFs are used by Ishibuchi *et al.* [77] to solve a biobjective problem $[f(\mathbf{x}), g(\mathbf{x})]^T$ by reformulating it as a four-objective one $[f(\mathbf{x}) - w \cdot g(\mathbf{x}), f(\mathbf{x}) + w \cdot g(\mathbf{x}), g(\mathbf{x}) - w \cdot f(\mathbf{x}), g(\mathbf{x}) + w \cdot f(\mathbf{x})]^T$, where w is a scalar parameter. Fig. 3(b) shows the relation between two original objectives $f(\mathbf{x})$ and $g(\mathbf{x})$, and four constructed objectives. When w is a small real number, $f(\mathbf{x}) - w \cdot g(\mathbf{x})$ and $f(\mathbf{x}) + w \cdot g(\mathbf{x})$ are similar to $f(\mathbf{x})$. Besides, $g(\mathbf{x}) - w \cdot f(\mathbf{x})$ and $g(\mathbf{x}) + w \cdot f(\mathbf{x})$ are similar to $g(\mathbf{x})$. The generated four-objective problem is used to solve the problem $[f(\mathbf{x}), g(\mathbf{x})]^T$ and obtains better results.

In terms of diversity, Ishibuchi *et al.* [36] showed that using different weight vectors, multiple runs of SOEA to solve $w_1f_1(\mathbf{x}) + w_2f_2(\mathbf{x})$ can outperform a single run of NSGA-II to solve $[f_1(\mathbf{x}), f_2(\mathbf{x})]^T$ on knapsack problems. In terms of convergence, using SOEA to solve $\sum_{i=1}^m w_i f_i(\mathbf{x})$ can outperform using NSGA-II to solve $\mathbf{F}(\mathbf{x}) = [f_1(\mathbf{x}), \ldots, f_m(\mathbf{x})]^T$ on many-objective knapsack problems. Furthermore, they used scalarizing function $\sum_{i=1}^m w_i f_i(\mathbf{x})$ to help NSGA-II solve the many-objective optimization problem $\mathbf{F}(\mathbf{x}) = [f_1(\mathbf{x}), \ldots, f_m(\mathbf{x})]^T$ [78]. A summary of MSF is presented in Table VI.

VI. Useful Suggestions, Advantages, and Disadvantages of Multiobjectivization

This section introduces the multiobjectivization from useful suggestions, advantages, and disadvantages.

A. Useful Suggestions on Multiobjectivization

Several guidelines for the proper design of multiobjectivization are concluded as follows. This kind of suggestion is valuable for the successful use of multiobjectivization on various optimization problems.

1) The PS of the reconstructed MOP should include at least one optimal solution to the original problem so that solving the reconstructed MOP can obtain the solution to the original problem [10], [13].
2) Multiobjectivization can be designed based on the likelihood of avoiding local optima in the main objective [13].
3) Using the static multiobjectivization may bias the search. A solution to this issue is to use dynamic multiobjectivization based on the collected feedback information during the evolution [3], [13], [14].
4) The best choice of multiobjectivization is dependent on decomposition/helper-objective assignment, the decomposition/helper-objective size, problem features, algorithm features, and so on [3] and [7].
5) Using a small number of helper-objectives/subobjectives is suggested by Jenson [3], [16] and Lochtefeld and Ciarallo [2], since using too many helper-objectives simultaneously makes it become a many-objective problem, which removes the selection pressure toward the main objective.
6) Helper-objectives/subobjectives with low complexity are preferred based on Occam's razor principle [3], [18].
7) Decomposed/segmented subobjectives should be conflicted with each other in some regions or have a certain complementary property [13]. It is better to create a broad PF, which helps escape from local optima [13].
8) Lochtefeld and Ciarallo [2], [7] recommended the balanced decomposition/segment, that is, dividing the components of the objective in relatively equal size or influence. A too-small component represents little problem sectioning, while a too-large component has highly complex and the same disadvantage as solving the original problem.

Additional suggestions for MVD are listed as follows.

1) Knowles *et al.* [1] suggested that the MVD separates out the conflicting aspects of the problem as independently as possible.
2) Each component of the original problem should be represented in the transformed MOP [7]. It is a necessary condition to ensure that the PS of the transformed MOP includes at least the best solution to the original problem.
3) Complementary decomposition was suggested and explored by Lochtefeld and Ciarallo [13].
4) It is important to introduce a strategy to keep the current best solution to the original problem in the MVD, since the current best solution may not always survive, if the nondominated size of all candidate solutions is larger than the population size [13].
5) Elementary landscape decomposition is a new and potential MVD strategy due to its theoretical basis [42].

Other suggestions for MHO include the following.

1) Domain knowledge can be introduced as helper-objectives, which can be seen as the preference of the user helping the search [18].
2) The original objective as the main interest of the DM needs to be prioritized in the search [18].
3) It is problem-dependent on how many and which helper-objectives should be used, as well as when to change the helper-objectives [3], [16].
4) On the one hand, helper-objectives should be chosen in conflict with the original objective, at least for some parts of the search space [3], [16]. Otherwise, the search would be the same as a single-criterion optimization [68]. On the other hand, helper-objectives must reflect some aspect of the problem which is helpful for the search [3], [13].

B. Advantages of Multiobjectivization

As the pioneering work, Knowles *et al.* [1] and Louis and Rawlins [34] have shown that the main advantage of multiobjectivization is reducing the number of local optima. On the one hand, the local optima of the original objective may not present in one or more of the new objectives in the transformed MOP [7]. On the other hand, dynamic multiobjectivization can help the algorithm escape local optima by swapping the search space among different multiobjectivizations [3], [16].

Multiobjectivization can also introduce some plateaus of incomparability solutions [8]. Based on the notion of inferiority in multiobjective optimization, incomparable moves can be acceptable. This helps prevent the algorithm from becoming trapped in local optima [1], [38]. These plateau regions were useful for destroying multimodal, deceptive, and/or neutral regions, enabling the escape of low-quality regions [15].

Besides, the multiobjectivization method can find a set of nondominated solutions instead of a single solution in the SOO method, and at no extra cost in function evaluations [39]. Thus, it allows the DM to choose the best tradeoff solutions [43]. Besides, the multiobjectivization method is less sensitive to the choice of the mutation probability due to its natural diversity keeping. Thus, less emphasis may be put on tuning the mutation rate [11], [20].

MVD can equalize the influence of different subproblems that range over vastly different scales and hence can swamp each other's search gradients [39]. When using MVD for a sum-of-parts problem, the decomposed subproblems has less local optima as they present in the original problem [2].

MHO methods may benefit from the emphasis on helper-objectives [39]. Moves that are incomparable will be considered acceptable when helper-objectives are added. Multiobjectivization allows the algorithm to escape local optima of the main objective by incomparable moves [3]. What is more, using the low-dimensional subproblem as the helper-objective may increase the evolvability of the original problem due to this low-dimensional subproblem searching a subspace. Domain or expert's knowledge can be introduced as the helper-objective to guide the search toward the optima faster [18].

C. Disadvantages of Multiobjectivization

Although multiobjectivization methods have achieved success in a lot of optimization problems. They also have negative effects [15]. The success of the multiobjectivization can be partly attributed to the introduction of plateaus that are expected to escape local optima. However, the presence of these plateaus may slow down convergence speed [39]. Besides, introducing plateaus of nondominated solutions may offer the chance to mislead the multiobjectivization to explore regions away from the global optimum [39].

The researchers have shown that the multiobjectivization method can make the search easier or harder [14], [38]. Moreover, it is problem-dependent on the optimal choice for how many and which decomposed subobjectives/helper-objectives should be used [3], [16]. The dynamic multiobjectivization method is a potential direction but in its infancy.

The current theoretical analyses mainly focus on runtime analyses. There is a lack of more general theoretical research except [7], [38]. Besides, most of the theoretical analyses are on combinatorial optimization problems. Theoretical research are scarce about continuous optimization, dynamic optimization, constrained optimization, multiobjective optimization.

Unreasonable decomposition may lead to that the PS of the MVD problem does not include the optimal solution to the original problem [10], [13]. The MVD formulation is less emphasis on the original objective explicitly. This may lead to the MVD method has the chance to discard the best found solution under the original objective [13]. If so, the MVD method may hamper the algorithmic performance [7]. Introducing a large archive is one way to address this drawback [39]. However, researchers have shown that the introduction of an archive can also make a problem easier [39] or harder [39], [49]. The reason is that introduction of the archive may influence which solutions to breed and survive [11], [49].

Using a component $f_1(\mathbf{x})$ of the original objective $f(\mathbf{x})$ as the helper-objective may weaken the effect of other components. More specifically, if a new best value occurs in the part $f(\mathbf{x}) - f_1(\mathbf{x})$ of an offspring solution and this fitness improvement appears together with larger fitness decreases in the helper-objective $f_1(\mathbf{x})$, this solution may not be rewarded by the helper-objective algorithm [7]. Using the low-dimensional subproblem as the helper-objective also has this risk.

Weight vectors need to be specified in advance for multiobjectivization via scalar functions and few studies are on multiobjectivization via segment methods.

VII. More Discussions on Multiobjectivization

Many other topics, including the extensions for various problem types, problem properties hard for SOO and easy for multiobjectivization, key parameters, theoretical analyses, test problems, and real-world applications of multiobjectivization are also worth discussing. Due to the space limit, the discussions on these topics are provided in the supplementary material.

VIII. Conclusion and Future Directions

With the increasing complexity of real-world optimization problems, multiobjectivization has attracted a lot of attention and achieved great success in solving complex optimization problems. This article attempts to provide a multifacet survey of multiobjectivization, covering three kinds of classical multiobjectivization methods, extensions for various problem types, problem features easy for multiobjectivization, theoretical analyses, test problems, and applications. The advantages, disadvantages, and guiding principles of multiobjectivization are also discussed. The survey is expected to offer a general overview of the development of multiobjectivization methods. Possible directions of multiobjectivization are discussed as follows.

1) How to multiobjectivize is usually problem-dependent. Open issues include but are not limited to which one is the best multiobjectivization, how many and which helper-objectives/subobjectives should be used, and when to change helper-objectives/subobjectives [2], [3], [16].
2) Static multiobjectivization may bais the search. It is desired to study how to multiobjectivize automatically [3], [9], [20]. Compared with static multiobjectivization, dynamic multiobjectivizations are less studied but yield promising benefits in promoting diversity and avoiding premature convergence [15].
3) Multiobjectivization can make a problem harder or easier, depending on the used multiobjectivization [63]. However, there are few suggestions on which kind of helper-objectives/subobjectives can facilitate/hinder the problem solved. It is interesting to design more universal/specialized multiobjectivization methods [1], [20].
4) Researching what constitutes a good multiobjectivization is another potential direction [16].
5) Diversity maintenance is one of the main benefits of multiobjectivization. However, multiobjectivization methods are less compared against other diversity preservation methods to show their competitiveness [15].
6) Multiobjectivization can find many other nondominated solutions, whose value is not utilized in the current studies [39]. How to use these nondominated solutions is a subject of future research, especially when the main objective is approximate rather than true [9].
7) It is still an open issue on the relationship between the number of local optima in the original problem and the number of local PS/plateaus under multiobjectivization.
8) The helper-objectives in the current studies are assigned the same importance as the original objective. However, keeping a preferred emphasis on the main objective is important but less studied, because the main objective highlights what really matters [20].
9) Balanced decomposition is less explicitly highlighted, but it is a hidden principle in many MVD methods [7]. A good decomposition needs to balance different objectives of MOP in constructing the MVD.

10) The current theoretical analyses pay more attention to the runtime analyses and combinatorial optimization. It is still scarce for more general theoretical studies on multiobjectivization except [7] and [38]. More theoretical results are needed on continuous optimization, dynamic optimization, constrained optimization, and so on.

References

[1] J. D. Knowles, R. A. Watson, and D. W. Corne, "Reducing local optima in single-objective problems by multi-objectivization," in *Proc. Int. Conf. Evol. Multi-Criterion Optim.*, 2001, pp. 269–283.

[2] D. F. Lochtefeld and F. W. Ciarallo, "Multiobjectivization via helper-objectives with the tunable objectives problem," *IEEE Trans. Evol. Comput.*, vol. 16, no. 3, pp. 373–390, Jun. 2012.

[3] M. T. Jensen, "Helper-objectives: Using multi-objective evolutionary algorithms for single-objective optimisation," *J. Math. Model. Algorithms*, vol. 3, no. 4, pp. 323–347, 2005.

[4] D. Brockhoff, T. Friedrich, N. Hebbinghaus, C. Klein, F. Neumann, and E. Zitzler, "Do additional objectives make a problem harder?" in *Proc. Genet. Evol. Comput. Conf.*, 2007, pp. 1–8.

[5] C. Segura, E. Segredo, Y. Gonzalez, and C. Leon, "Multiobjectivisation of the antenna positioning problem," in *Proc. Int. Symp. Distrib. Comput. Artif. Intell.*, 2011, pp. 319–328.

[6] H. Ishibuchi and Y. Nojima, "Optimization of scalarizing functions through evolutionary multiobjective optimization," in *Evolutionary Multi-Criterion Optimization*. Heidelberg, Germany: Springer, 2007, pp. 51–65.

[7] D. F. Lochtefeld and F. W. Ciarallo, "Multi-objectivization via decomposition: An analysis of helper-objectives and complete decomposition," *Eur. J. Oper. Res.*, vol. 243, no. 2, pp. 395–404, 2015.

[8] M. Garza-Fabre, G. Toscano-Pulido, and E. Rodriguez-Tello, "Multi-objectivization, fitness landscape transformation and search performance: A case of study on the HP model for protein structure prediction," *Eur. J. Oper. Res.*, vol. 243, no. 2, pp. 405–422, 2015.

[9] T. Brys, A. Harutyunyan, P. Vrancx, A. Nowe, and M. E. Taylor, "Multi-objectivization and ensembles of shapings in reinforcement learning," *Neurocomputing*, vol. 263, pp. 48–59, Nov. 2017.

[10] F. Neumann and I. Wegener, "Can single-objective optimization profit from multiobjective optimization?" in *Proc. Multiobjective Problem Solving Nat. Concepts Appl.*, 2008, pp. 115–130.

[11] D. F. Lochtefeld and F. W. Ciarallo, "Deterministic helper-objective sequences applied to job-shop scheduling," in *Proc. Genet. Evol. Comput. Conf.*, 2010, pp. 431–438.

[12] E. Segredo, C. Segura, and C. Leon, "Memetic algorithms and hyperheuristics applied to a multiobjectivised two-dimensional packing problem," *Appl. Soft Comput.*, vol. 58, no. 4, pp. 769–794, 2014.

[13] D. F. Lochtefeld and F. W. Ciarallo, "An analysis of decomposition approaches in multi-objectivization via segmentation," *Appl. Soft Comput.*, vol. 18, pp. 209–222, May 2014.

[14] M. Jahne, X. Li, and J. Branke, "Evolutionary algorithms and multi-objectivization for the travelling salesman problem," in *Proc. Conf. Genet. Evol. Comput.*, 2009, pp. 595–602.

[15] C. Segura, C. A. C. Coello, G. Miranda, and C. Leín, "Using multi-objective evolutionary algorithms for single-objective constrained and unconstrained optimization," *Ann. Oper. Res.*, vol. 240, no. 1, pp. 217–250, 2016.

[16] M. T. Jensen, "Guiding single-objective optimization using multi-objective methods," in *Proc. EvoWorkshops Conf.*, 2003, pp. 268–279.

[17] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints," *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 577–601, Aug. 2014.

[18] A. Syberfeldt and J. Rogstrom, "A two-step multi-objectivization method for improved evolutionary optimization of industrial problems," *Appl. Soft Comput.*, vol. 64, pp. 331–340, Mar. 2018.

[19] H. Ishibuchi, Y. Hitotsuyanagi, Y. Nojima, and Y. Nojima, "Multiobjectivization from two objectives to four objectives in evolutionary multi-objective optimization algorithms," in *Proc. 2nd World Congr. Nat. Biol. Inspired Comput.*, 2010, pp. 1143–1150.

[20] D. Lochtefeld and F. W. Ciarallo, "Helper-objective optimization strategies for the job-shop scheduling problem," *Appl. Soft Comput.*, vol. 11, no. 6, pp. 4161–4174, 2011.

[21] C. Segura, C. A. C. Coello, G. Miranda, and C. Leon, "Using multi-objective evolutionary algorithms for single-objective optimization," *4OR*, vol. 11, no. 3, pp. 201–228, 2013.

[22] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*. New York, NY, USA: Wiley, 2001.

[23] C. A. C. Coello, "Evolutionary multi-objective optimization: A historical view of the field," *IEEE Comput. Intell. Mag.*, vol. 1, no. 1, pp. 28–36, Feb. 2006.

[24] X. Ma *et al.*, "Enhanced multifactorial evolutionary algorithm with meme helper-tasks," *IEEE Trans. Cybern.*, early access, Feb. 10, 2021, doi: 10.1109/TCYB.2021.3050516.

[25] A. Zhou, B. Qu, H. Li, S. Zhao, P. N. Suganthan, and Q. Zhang, "Multiobjective evolutionary algorithms: A survey of the state of the art," *Swarm Evol. Comput.*, vol. 1, no. 1, pp. 32–49, 2011.

[26] J. Lin, H. L. Liu, K. C. Tan, and F. Gu, "An effective knowledge transfer approach for multiobjective multitasking optimization," *IEEE Trans. Cybern.*, vol. 51, no. 6, pp. 3238–3248, Jun. 2021.

[27] M. Gong, X. Jiang, H. Li, and K. C. Tan, "Multiobjective sparse non-negative matrix factorization," *IEEE Trans. Cybern.*, vol. 49, no. 8, pp. 2941–2954, Aug. 2019.

[28] Z. Liang, K. Hu, X. Ma, and Z. Zhu, "A many-objective evolutionary algorithm based on a two-round selection strategy," *IEEE Trans. Cybern.*, vol. 51, no. 3, pp. 1417–1429, Mar. 2021.

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

[30] L. Ma *et al.*, "Learning to optimize: Reference vector reinforcement learning adaption to constrained many-objective optimization of industrial copper burdening system," *IEEE Trans. Cybern.*, early access, Jul. 14, 2021, doi: 10.1109/TCYB.2021.3086501.

[31] D. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Reading, MA, USA: Addison-Wesley, 1989.

[32] R. Salomon, "Re-evaluating genetic algorithm performance under coordinate rotation of benchmark functions—A survey of some theoretical and practical aspects of genetic algorithms," *BioSystems*, vol. 39, no. 3, pp. 263–278, 1995.

[33] D. Yazdani, M. N. Omidvar, R. Cheng, J. Branke, and X. Yao, "Benchmarking continuous dynamic optimization: Survey and generalized test suite," *IEEE Trans. Cybern.*, early access, Aug. 14, 2020, doi: 10.1109/TCYB.2020.3011828.

[34] S. J. Louis and G. J. E. Rawlins, "Pareto optimality, GA-easiness and deception," in *Proc. 5th Int. Conf. Genet. Algorithms*, 1993, pp. 1–6.

[35] J. Shi, J. Sun, and Q. Zhang, "Multi-objectivizing sum-of-the-parts combinatorial optimization problems by random objective decomposition," 2019. *arXiv:1911.04658*.

[36] H. Ishibuchi, Y. Nojima, and T. Doi, "Comparison between single-objective and multi-objective genetic algorithms: Performance comparison and performance measures," in *Proc. IEEE Congr. Evol. Comput.*, 2006, pp. 1143–1150.

[37] I. V. Silva, N. C. Cortes, G. T. Pulido, and L. G. de la Fraga, "Optimal triangulation in 3D computer vision using a multi-objective evolutionary algorithm," in *Proc. Workshops Appl. Evol. Comput.*, 2007, pp. 493–502.

[38] J. Handl, S. C. Lovell, and J. Knowles, "Multiobjectivization by decomposition of scalar cost functions," in *Proc. 10th Int. Conf. Parallel Problem Solving Nat.*, 2008, pp. 349–366.

[39] J. Handl, S. Lovell, and J. Knowles, "Investigations into the effect of multiobjectivization in protein structure prediction," in *Proc. Int. Conf. Parallel Problem Solving Nat.*, 2008, pp. 1–11.

[40] M. Gong, J. Liu, H. Li, Q. Cai, and L. Su, "A multiobjective sparse feature learning model for deep neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 12, pp. 3263–3277, Dec. 2015.

[41] H. Li, J. Sun, M. Wang, and Q. Zhang, "MOEA/D with chain-based random local search for sparse optimization," *Soft Comput.*, vol. 22, pp. 7087–7102, Sep. 2018.

[42] J. Ceberio, B. Calvo, A. Mendiburu, and J. A. Lozano, "Multi-objectivising combinatorial optimisation problems by means of elementary landscape decompositions," *Evol. Comput.*, vol. 27, no. 2, pp. 291–311, 2019.

[43] A. J. Nebro, E. Alba, G. Molina, J. F. Chicano, and J. J. Durillo, "Optimal antenna placement using a new multi-objective CHC algorithm," in *Proc. Genet. Evol. Comput. Conf.*, 2007, pp. 876–883.

[44] A. W. Churchill, P. Husbands, and A. Philippides, "Multi-objectivization of the tool selection problem on a budget of evaluations," in *Proc. Evol. Multi-Criterion Optim.*, 2013, pp. 600–614.

[45] M. Soltan, P. Derakhshanfar, A. Panichella, X. Devroey, A. Zaidman, and A. van Deursen, "Single-objective versus multi-objectivized optimization for evolutionary crash reproduction," in *Proc. Int. Symp. Search Based Softw. Eng.*, 2018, pp. 325–340.

[46] C. A. C. Coello and A. H. Aguirre, "Design of combinational logic circuits through an evolutionary multiobjective optimization approach," *Artif. Intell. Eng. Design Anal. Manuf.*, vol. 16, no. 1, pp. 39–53, 2002.

[47] J. Scharnow, K. Tinnefeld, and I. Wegener, "Fitness landscapes based on sorting and shortest path problems," in *Parallel Problem Solving from Nature*. Heidelberg, Germany: Springer, 2002, pp. 54–63.

[48] J. Scharnow, K. L. Tinnefeld, and I. Wegener, "The analysis of evolutionary algorithms on sorting and shortest paths problems," *J. Math. Model. Algorithms*, vol. 3, no. 4, pp. 349–366, 2004.

[49] M. Garza-Fabre, E. Rodriguez-Tello, and G. Toscano-Pulido, "Multiobjectivizing the HP model for protein structure prediction," in *Proc. 12th Eur. Conf. Evol. Comput. Combinatorial Optim.*, 2012, pp. 1–12.

[50] M. Garza-Fabre, E. Rodriguez-Tello, and P. G. Toscano, "An improved multiobjectivization strategy for HP model-based protein structure prediction," in *Proc. Int. Conf. Parallel Problem Solving Nat.*, 2012, pp. 241–248.

[51] H. A. Abbass and K. Deb, "Searching under multi-evolutionary pressures," in *Proc. 2nd Int. Conf. Evol. Multi-Criterion Optim.*, 2003, pp. 391–404.

[52] S. Watanabe and K. Sakakibara, "Multi-objective approaches in a single-objective optimization environment," in *Proc. IEEE Congr. Evol. Comput.*, 2005, pp. 1–8.

[53] M. Buzdalov and A. Buzdalova, "Onemax helps optimizing XDIVK: Theoretical runtime analysis for RLS and EA+RL," in *Proc. Genet. Evol. Comput. Conf.*, 2014, pp. 1–2.

[54] S. A. Thomas and Y. Jin, "Single and multi-objective in silico evolution of tunable genetic oscillators," in *Evolutionary Multi-Criterion Optimization*. Heidelberg, Germany: Springer, 2013, pp. 696–709.

[55] T. Friedrich, N. Hebbinghaus, and F. Neumann, "Plateaus can be harder in multi-objective optimization," in *Proc. IEEE Congr. Evol. Comput.*, 2007, pp. 2622–2629.

[56] T. Friedrich, J. He, N. Hebbinghaus, F. Neumann, and C. Witt, "Approximating covering problems by randomized search heuristics using multi-objective models," in *Proc. Genet. Evol. Comput. Conf.*, 2010, pp. 617–633.

[57] J. Yao, N. Kharma, and P. Grogono, "Bi-objective multipopulation genetic algorithm for multimodal function optimization," *IEEE Trans. Evol. Comput.*, vol. 14, no. 1, pp. 80–102, Feb. 2010.

[58] A. Alsheddy, "A penalty-based multi-objectivization approach for single objective optimization," *Inf. Sci.*, vols. 442–443, pp. 1–17, 2018.

[59] H. Ishibuchi, H. Masuda, and Y. Nojima, "Meta-level multi-objective formulations of set optimization for multi-objective optimization problems: Multi-reference point approach to hypervolume maximization," in *Proc. Genet. Evol. Comput. Conf.*, 2014, pp. 89–90.

[60] A. Bucci and J. B. Pollack, "On identifying global optima in cooperative coevolution," in *Proc. Genet. Evolut. Comput. Conf.*, 2005, pp. 539–544.

[61] J. Noble and R. A. Watson, "Pareto coevolution: Using performance against coevolved opponents in a game as dimensions for Pareto selection," in *Proc. Genet. Evol. Comput. Conf.*, 2001, pp. 493–500.

[62] M. Wright, "Subcost-guided search-experiments with timetabling problems," *J. Heurist.*, vol. 7, no. 3, pp. 251–260, 2001.

[63] D. Brockhoff, T. Friedrich, N. Hebbinghaus, C. Klein, F. Neumann, and E. Zitzler, "On the effects of adding objectives to plateau functions," *IEEE Trans. Evol. Comput.*, vol. 13, no. 3, pp. 591–603, Jun. 2009.

[64] D. Kim, "Minimizing structural risk on decision tree classification," in *Multi-Objective Machine Learning*. Heidelberg, Germany: Springer, 2006, pp. 241–260.

[65] E. D. d. Jong, R. A. Watson, and J. B. Pollack, "Reducing bloat and promoting diversity using multi-objective methods," in *Proc. Genet. Evol. Comput. Conf.*, 2001, pp. 11–18.

[66] S. Bleuler, M. Brack, L. Thiele, and C. Zitzler, "Multiobjective genetic programming: Reducing bloat using SPEA2," in *Proc. IEEE Congr. Evol. Comput.*, 2001, pp. 536–543.

[67] M. Gong, M. Zhang, and Y. Yuan, "Unsupervised band selection based on evolutionary multiobjective optimization for hyperspectral images," *IEEE Trans. Geosci. Remote Sens.*, vol. 1, no. 54, pp. 544–557, Jan. 2016.

[68] D. Greiner, J. M. Emperador, G. Winter, and B. Galvan, *Improving Computational Mechanics Optimum Design Using Helper Objectives: An Application in Frame Bar Structures* (Lecture Notes in Computer Science 4403). Heidelberg, Germany: Springer, 2007, pp. 575–589.

[69] S. Watanabe and K. Sakakibara, "A multiobjectivization approach for vehicle routing problems," in *Proc. Int. Conf. Evol. Multi-Criterion Optim.*, 2006, pp. 142–155.

[70] F. Neumann and I. Wegener, "Minimum spanning trees made easier via multi-objective optimization," *Natural Comput.*, vol. 5, no. 3, pp. 305–319, 2005.

[71] D. Sharma, K. Deb, and N. N. Kishore, "Customized evolutionary optimization procedure for generating minimum weight compliant mechanisms," *Eng. Optim.*, vol. 46, no. 1, pp. 39–60, 2014.

[72] A. Trivedi, D. Sharma, and D. Srinivasan, "Multi-objectivization of short-term unit commitment under uncertainty using evolutionary algorithm," in *Proc. IEEE Congr. Evol. Comput.*, 2012, pp. 1–8.

[73] J. Durillo and T. Fahringer, "From single- to multi-objective auto-tuning of programs: Advantages and implications," *Sci. Program.*, vol. 22, pp. 285–297, Jan. 2014.

[74] A. Buzdalova and M. Buzdalov, "A new algorithm for adaptive online selection of auxiliary objectives," in *Proc. Int. Conf. Mach. Learn. Appl.*, 2014, pp. 584–588.

[75] M. Buzdalov and A. Buzdalova, "Adaptive selection of helper-objectives for test case generation," in *Proc. IEEE Congr. Evol. Comput.*, 2013, pp. 245–250.

[76] A. Buzdalova and M. Buzdalov, "Adaptive selection of helper-objectives with reinforcement learning," in *Proc. Int. Conf. Mach. Learn. Appl.*, 2012, pp. 217–250.

[77] H. Ishibuchi, Y. Sakane, N. Tsukamoto, and Y. Nojima, "Simultaneous use of different scalarizing functions in MOEA/D," in *Proc. Genet. Evol. Comput. Conf.*, 2010, pp. 519–526.

[78] H. Ishibuchi, T. Doi, and Y. Nojima, "Incorporation of scalarizing fitness functions into evolutionary multiobjective optimization algorithms," in *Proc. 9th Int. Conf. Parallel Problem Solving Nat.*, 2006, pp. 493–502.

Xiaoliang Ma (Member, IEEE) received the B.S. degree in computing computer science and technology from Zhejiang Normal University, Jinhua, China, in 2006, and the Ph.D. degree from the School of Computing, Xidian University, Xi'an, China, in 2014.

He is currently an Assistant Professor with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China. His research interests include evolutionary computation, multiobjective optimization, and cooperative coevolution.

Zhitao Huang received the B.S. degree in computer science and technology from Shenzhen University, Shenzhen, China, in 2019, where he is currently pursuing the M.S. degree with the College of Computer and Software.

His current research interests include evolutionary computation, multiobjective optimization, and various heuristic algorithms.

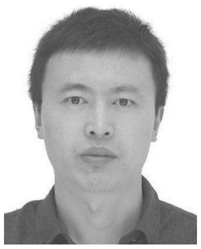
Xiaodong Li (Fellow, IEEE) received the B.Sc. degree from Xidian University, Xi'an, China, in 1988, the master's degree in 1992, and the Ph.D. degree in information science from the University of Otago, Dunedin, New Zealand, in 1998.

He is a Professor with the School of Science (Computer Science and Software Engineering), RMIT University, Melbourne, VIC, Australia. His research interests include machine learning, evolutionary computation, neural networks, data analytics, multiobjective optimization, multimodal optimization, and swarm intelligence.

Prof. Li was a recipient of the 2013 ACM SIGEVO Impact Award and the 2017 IEEE CIS IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION Outstanding Paper Award. He serves as an Associate Editor of the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, *Swarm Intelligence* (Springer), and *International Journal of Swarm Intelligence Research*. He is a Founding Member of IEEE CIS Task Force on Swarm Intelligence, the Vice-Chair of IEEE Task Force on Multimodal Optimization, and the Former Chair of IEEE CIS Task Force on Large Scale Global Optimization.

Yutao Qi received the B.Sc. degree in software engineering, the M.Sc. degree in computer science and technology, and the Ph.D. degree in pattern recognition and intelligent system from Xidian University, Xi'an, China, in 2003, 2006, and 2008, respectively.

He is a Professor with the School of Computer Science and Technology, Xidian University. His main research interests include evolutionary computation, artificial immune system, and engineering optimization methods.



Lei Wang received the Ph.D. degree from Xidian University, Xi'an, China, in 2010.

He was worked with Huawei Technologies Company, Ltd., Shenzhen, China, from 2011 to 2012. He is currently an Associate Professor with the Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen. His research interests include image transforms, machine learning, computer vision, visual semantic understanding, video analysis, and deep learning.

Zexuan Zhu (Senior Member, IEEE) received the B.S. degree in computer science and technology from Fudan University, Shanghai, China, in 2003, and the Ph.D. degree in computer engineering from Nanyang Technological University, Singapore, in 2008.

He is currently a Professor with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China. His research interests include computational intelligence, machine learning, and bioinformatics.

Prof. Zhu is an Associate Editor of IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION and IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTATIONAL INTELLIGENCE.