

# A Survey on Learnable Evolutionary Algorithms for Scalable Multiobjective Optimization

Songbai Liu, *Member, IEEE*, Qiuzhen Lin, *Member, IEEE*, Jianqiang Li, *Member, IEEE*,  
and Kay Chen Tan, *Fellow, IEEE*

**Abstract**—Recent decades have witnessed great advancements in multiobjective evolutionary algorithms (MOEAs) for multi-objective optimization problems (MOPs). However, these progressively improved MOEAs have not necessarily been equipped with scalable and learnable problem-solving strategies for new and grand challenges brought by the scaling-up MOPs with continuously increasing complexity from diverse aspects, mainly including expensive cost of function evaluations, many objectives, large-scale search space, time-varying environments, and multi-task. Under different scenarios, divergent thinking is required in designing new powerful MOEAs for solving them effectively. In this context, research studies on learnable MOEAs with machine learning techniques have received extensive attention in the field of evolutionary computation. This paper begins with a general taxonomy of scaling-up MOPs and learnable MOEAs, followed by an analysis of the challenges that these MOPs pose to traditional MOEAs. Then, we synthetically overview recent advances of learnable MOEAs in solving various scaling-up MOPs, focusing primarily on four attractive directions (i.e., learnable evolutionary discriminators for environmental selection, learnable evolutionary generators for reproduction, learnable evolutionary evaluators for function evaluations, and learnable evolutionary transfer modules for sharing or reusing optimization experience). The insight of learnable MOEAs is offered to readers as a reference to the general track of the efforts in this field.

**Index Terms**—Learnable Evolutionary Algorithms, Machine Learning, Scalable Multiobjective Optimization.

## I. INTRODUCTION

**E**VOLUTIONARY algorithms (EAs) are generally regarded as nondeterministic problem solvers inspired by biological evolution mechanisms such as natural selection and genetic variation. They can find a set of approximately optimal solutions for the problem without requiring domain-specific information such as continuous, differentiable, and unimodal functions. Using heuristic random search mechanisms, EAs have a strong ability to cope with uncertainty, which are suitable for

finding global and robust optimal solutions [1]. In addition, EAs with population-based search are inherently capable of parallel computing [2]. More importantly, most EAs can obtain a set of solutions in a single run, so they have advantages in solving multiobjective optimization problems (MOPs) compared with mathematical programming approaches [3]. Thus, EAs are widely used in many real-world applications to solve specific MOPs that have plagued traditional mathematical optimizers [4, 5]. Especially, multiobjective EAs (MOEAs) have been applied to tackle MOPs in other research areas, e.g., evolutionary machine learning (ML) [6, 7], evolutionary data mining [8, 9], evolutionary automatic network design [10], and evolutionary transfer learning [11]. This progress in various areas, along with the increasing complexities faced by engineering and science, suggests that MOEAs are effective for problem-solving in many applications.

In general, an MOEA starts with an initial parent population of randomly sampled solutions. Then, it reproduces a population of offspring solutions (with some features inherited from their parents) by a generative or reproductive model. This model, termed evolutionary generator, is mainly equipped with some genetic operators [12], including mating selection, crossover, mutation, etc. All newly generated solutions need to go through a function evaluator to calculate the value of each of their objectives. After that, the combined population of parent and offspring is inputted into a discriminative (or selective) model. This model, referred to as evolutionary discriminator, outputs only the elite solutions discriminated to be fitter for surviving in the next generation (i.e., the natural or environmental selection based on the “survival of the fittest” principle [13]). From the perspective of ML as shown in Fig. 1, an MOEA’s generator aims to reproduce offspring with superior qualities over their parents by evolution, while its discriminator aims to distinguish the qualities between parent and offspring solutions after the evaluator, followed by selecting a certain number of elites to the next generation. Together, they form the driving force for searching diverse and progressively converged solutions to approximate the unknown optima of the given MOPs [14].

After decades of development, MOEAs should have grown into sophisticated, innovative, and creative solvers for various MOPs, especially from the perspective of biological evolution [15]. However, most existing MOEAs often lack the flexibility, plasticity, and scalability to effectively respond to major challenges brought by scaling-up MOPs. These problems grow

Manuscript received August xx, 2022; revised October 2022, and January 2023; accepted February 2023. This work is partially supported by the National Natural Science Foundation of China (NSFC) under No. U21A20512, and in part by the Research Grants Council of the Hong Kong SAR under Grant PolyU11211521 and Grant PolyU15218622, and in part by The Hong Kong Polytechnic University (Project No.: 1-ZE0C), and in part by Shenzhen Science and Technology Program under Grants JCYJ20220531101411027, JCYJ20190808164211203 and R2020A045, and in part by Natural Science Foundation of Guangdong Province under Grant No. 2023A1515011238. (Corresponding authors: Qiuzhen Lin and Kay Chen Tan)

S.B. Liu, Q.Z. Lin, and J.Q. Li are with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China (e-mail: songbai@szu.edu.cn).

K. C. Tan is with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong SAR (e-mail: kctan@polyu.edu.hk).

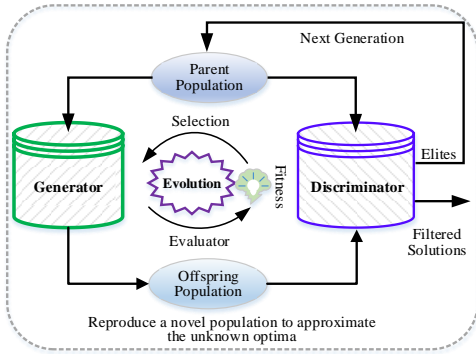


Fig. 1. The general flow of an MOEA from the perspective of ML.

in complexity as they are scaled up from diverse aspects, mainly including the expensive cost of function evaluations [16], many objectives [17], large-scale decision variables [18], time-varying environments [19], and multitasking [20]. The potential obstacles that degrade the performance of traditional MOEAs in solving these scaling-up MOPs are elaborately summarized in Section II.B. Moreover, with the rapid increase in the scale and complexity of today's engineering designs and available data, existing problem-solving systems cannot comprehend the "big dimensionality" spaces [21] or adaptively cope with all potential challenges brought by "big complex optimization problems" [22]. It is becoming pressingly demanded to design more intelligent and powerful MOEAs for tackling increasingly complex MOPs. Evolutionary optimization and incremental learning are two instinctual ways for humans to improve their problem-solving abilities. Thus, solving the MOPs only by evolution in traditional MOEAs may be inadequate and inefficient, especially when solving those scaling-up MOPs with diverse complexities. As stated in [15], biological evolution may take thousands of years to optimize a species, while cumulative learning can help speed up the optimization process dramatically. Inspired by this, an insightful research question was born, i.e., can ML be incorporated into MOEAs (i.e., learnable MOEAs) to form a more powerful optimizer that solves scaling-up MOPs efficiently?

In this context, research studies on learnable MOEAs have received extensive attention in the field of evolutionary computation [23]. ML techniques are applied to assist and catalyze the evolutionary modules (i.e., generator, discriminator, and evaluator) in learnable MOEAs. Particularly, the generator of an MOEA can generate a huge volume of data (feasible solutions) by iteratively searching in the variable space. Systematic analysis of these data by ML techniques is helpful to better understand the search behavior and improve its future search capability [24]. Exploring along promising directions learned in the search space, the MOEA's generator can effectively find solutions with great potential [25]. Through online predicting Pareto Front (PF) shapes, the MOEA's discriminator will be more competent to adaptively filter out poorly performing solutions when solving MOPs with various irregular PFs [26]. The objective and search spaces can be simplified with the help of dimension reduction and spatial transformation techniques before being tackled [27]. Reinforcement learning (RL) can

be employed to determine appropriate evolutionary operators (i.e., actions) given any parent (i.e., state), which can guide the generator to reproduce high-quality offspring [28]. Domain adaptation can be used to learn the domain-invariant feature representations between different MOPs and then analyze their distribution divergence, which guides the knowledge transfer when solving them sequentially or simultaneously [29].

Therefore, during the evolutionary process, MOEAs incessantly produce diverse data and information, which can be learned using ML to extract and purify knowledge for further enhancing their performance. It can be predicted that learnable MOEAs will play an essential role in handling complex MOPs in the future development of MOEAs. A closer inspection of the most reputed scientific databases also unveils those efforts carried out in learnable MOEAs are exponentially growing in recent years. This upsurge of studies demands insightful reference material to 1) summarize related achievements so far, 2) detect and analyze research trends, 3) discuss current limitations and perform a profound reflection, and 4) prescribe some interesting future research directions. This is the rationale for this survey. Thus, a comprehensive review of existing efforts on learnable MOEAs for scaling-up MOPs is proposed from the following five aspects:

- 1) General definition and taxonomy of scaling-up MOPs and learnable MOEAs;
- 2) Efforts on learnable evolutionary discriminators for handling the MOPs with a scaling-up objective space, i.e., many-objective optimization problems (MaOPs);
- 3) Efforts on learnable evolutionary generators for solving the MOPs with a scaling-up search space, i.e., large-scale MOPs (LMOPs);
- 4) Efforts on learnable evaluators to alleviate the challenges posed by MOPs with the scaling-up cost of function evaluations, i.e., expensive MOPs (EMOPs);
- 5) Efforts on learnable evolutionary transfer modules for finding a shortcut that can improve the efficacy when optimizing a scaling-up number of different MOPs sequentially or simultaneously, i.e., sequential MOPs (SMOPs) and multitasking MOPs (MMOPs).

It is worth pointing out that many related submissions have been made available publicly, but there is no comprehensive survey of the literature on learnable MOEAs. Although a review on evolutionary computation plus ML has been made in [23], it mainly focuses on reviewing different ML methods applied to assist swarm intelligence algorithms, rather than concentrating on the learnable MOEAs. Besides, most of the references in [23] are pre-2012 and do not include an update on the papers published during the past ten years for learnable MOEAs. This paper presents a survey involving a myriad of new learnable MOEA studies, with the expectation to inspire some new ideas for promoting the development of MOEA in solving complex MOPs.

The rest of this paper is organized as follows. We first provide the general definition and taxonomy of scalable and learnable evolutionary multiobjective optimization (EMO) in Section II. Then, we introduce the existing efforts with respect to learnable discriminators, learnable generators, learnable

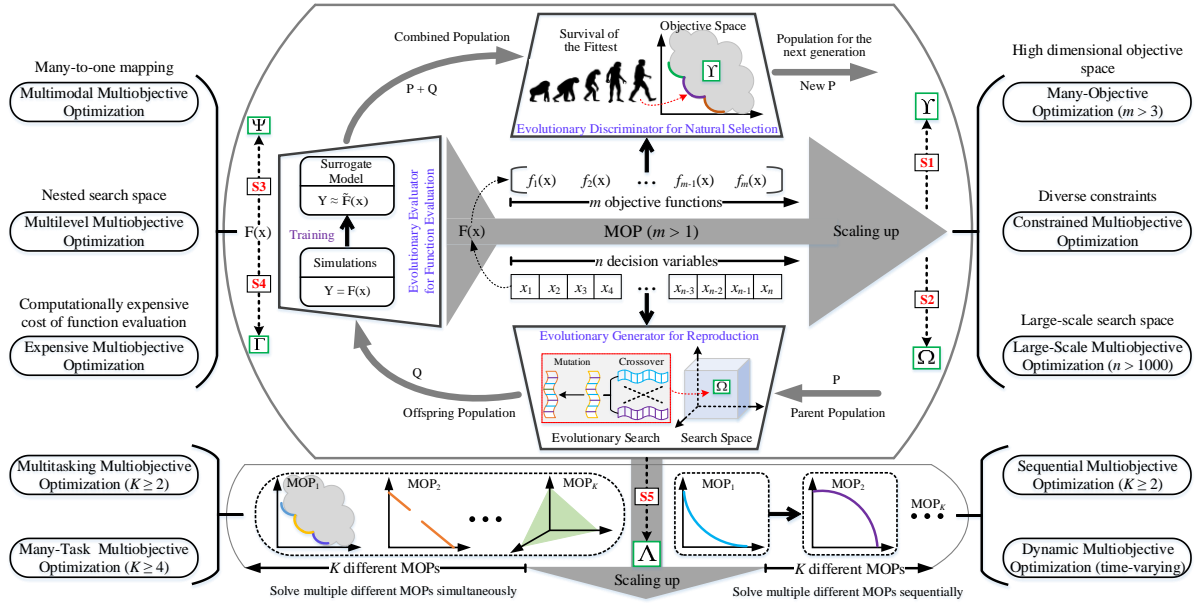


Fig. 2. Taxonomy of scaling-up MOPs according to the struct  $S_{MOP}$  from its five scalable components, including S1: objective space  $\Upsilon$ , S2: search space  $\Omega$ , S3: mapping diagram  $\Psi$ , S4: computational cost  $\Gamma$  for function evaluation, and S5: problem domain  $\Lambda$ .

evaluators, and learnable transfer modules in Sections III-VI. Finally, we conclude this paper and provide some potential research directions in Sections VII-VIII.

## II. SCALABLE AND LEARNABLE EMO

This section briefly introduces the preliminaries related to scalable EMO, including the general definition and taxonomy of scaling-up MOPs and the challenges they bring. Accordingly, we present the taxonomy of learnable MOEAs.

### A. Scalable Multiobjective Optimization Problems

An MOP comprises multiple objective functions to be minimized simultaneously, which can be formulated as follows:

$$\begin{aligned} \text{Minimize } F(x) &= (f_1(x), \dots, f_m(x)), \\ \text{s.t. } x &\in \Omega, F(x) \in \Upsilon \end{aligned} \quad (1)$$

where  $x = (x_1, \dots, x_n)$  is a solution vector with  $n$  often inter-related variables from the search space  $\Omega$ , and  $F(x)$  defines  $m$  often mutually conflicted functions  $f_1(x), \dots, f_m(x)$  in the objective space  $\Upsilon$ , none of which can be preferred over others ( $m \geq 2$ ). Therefore, without additional subjective preference, we cannot find a single optimal solution to simultaneously optimize all objectives of an MOP, but a (possibly infinite) set of Pareto optimal solutions, called the Pareto set (PS). The mapping of PS in the objective space is called PF. Thus, the goal of solving an MOP by MOEAs may be to find a representative set of solutions that can evenly and closely approximate the PS/PF [30], or to find a single solution that satisfies the subjective preferences of a decision-maker [31]. In this paper, a struct  $S_{MOP}$ , including five scalable components, is defined to express the scalability of MOPs, as follows:

$$S_{MOP} = (\Omega, \Upsilon, \Psi, \Gamma, \Lambda) \quad (2)$$

where  $\Upsilon$  and  $\Omega$  respectively represent the objective space and the search space of an  $MOP \in \Lambda$ ;  $\Psi$  is the mapping diagram of  $x \rightarrow F(x)$  in (1) for this MOP;  $\Gamma$  denotes the computational cost of evaluating  $F(x)$ ; and  $\Lambda$  indicates the problem domain including  $K$  different MOPs to be solved,  $K \geq 1$ . According to this struct  $S_{MOP}$ , the taxonomy of scaling-up MOPs is illustrated in Fig. 2. The details are elaborated below.

From the aspect of scalable  $\Upsilon$ , an MOP with  $m = 2$  or 3 can be scaled up to become an MaOP with  $m \geq 4$ . Since constraints can be reformulated as optimization objectives, the MOP with scaling-up constraints can be implicitly regarded as an MaOP. Researchers interested in constrained EMO can refer to its latest survey [32]. Considering the scalable  $\Omega$ , an MOP with a small-scale  $\Omega$  can be switched to an LMOP with a large-scale  $\Omega$ , in which its dimensionality  $n$  scales up to more than one thousand, i.e.,  $n \geq 1000$  [33].

Regarding the scalable  $\Psi$ , an MOP with a one-to-one mapping can be scaled up to become a multimodal MOP with a many-to-one mapping, in which multiple diverse optimal variable vectors in  $\Omega$  may share the nearly same objective vector in  $\Upsilon$ . Besides, most MOPs follow direct mapping, i.e., the vector  $x$  is directly paired to the function vector  $F(x)$  without involving any nested mapping. However, such direct mapping does not reflect the complexity level of practical problems, where the mapping may be modulated through multiple levels of indirection, e.g., bi-level MOP with a nested search space. Interested readers can refer to the latest surveys [34] and [35] respectively for multimodal EMO and bi-level EMO. Since the  $F(x)$  may be unexpressed and noisy in practice, its evaluation will become more and more computationally expensive as the fidelity requirement increases [36]. Thus, from the aspect of implicitly scalable  $\Gamma$ , an inexpensive MOP can be scaled up to become an expensive MOP (EMOP), which requires prohibitive computational cost for function evaluations.

TABLE I  
SUMMARIZATION OF THE BOTTLENECK FOR TRADITIONAL MOEAS IN SOLVING SCALING-UP MOPS

Aspects	Scale-up MOPs	Main modules	Basic strategies	Challenges or limitations
S1 – $\Upsilon$ : objective space	◊ MaOPs (many-objective)	✓ Discriminator; //outputs survival solutions by filtering poorly performed solutions from the inputted population	Selection strategies: ⊕ Pareto-based [41]; ⊕ indicator-based [42]; ⊕ decomposition-based [43]; ⊕ their hybridization [44];	⊙ insufficient selection pressure for Pareto-based strategies (mutually non-dominated solutions) [45]; ⊙ efficacy of decomposition-based strategies deteriorates severely as pre-specified weights cannot match irregular PFs well [46]; ⊙ cost of computing performance indicators grows exponentially as objective increases [47].
S2 – $\Omega$ : search space	◊ LMOPs (large-scale)	✓ Generator; //outputs offspring by finding promising solutions in search space based on the inputted parents	Search strategies: ⊕ mutation; ⊕ crossover; ⊕ particle swarm optimizer; ⊕ differential evolution; ⊕ evolution strategy [48]; ⊕ etc.	⊙ search space expands exponentially, requiring generator with growing search capability [49]; ⊙ interaction between variables are more intricate; ⊙ contribution of variables is more imbalanced; ⊙ overlap between shared variables that define different objectives becomes more disorganized [50]; ⊙ overly-increasing number of variables will inevitably make function evaluation expensive [51].
S3 – $\Psi$ : mapping diagram	◊ MMMOPs (multimodal) ◊ BMOPs (bi-level)	✓ Discriminator; ✓ Generator;	Integrated strategies: ⊕ diversity strategies cover objective & search space; ⊕ search strategies cover low- & up-level variables.	⊙ difficult to maintain diversity in both objective & search space of multimodal cases [34]; ⊙ difficult to find Pareto solutions with both optimal upper- & lower-level variables [35].
S4 – $\Gamma$ : computation cost for function evaluation	◊ EMOPs (expensive)	✓ Evaluator; //outputs objective function values or fitness values for each inputted solution	Evaluation strategies: ⊕ evaluate objectives from given expressions; ⊕ predict objectives by surrogates learned from simulation data when they are unexpressed or noisy.	⊙ require prohibitive computing cost by evaluating objective functions directly [52]; ⊙ construct proper surrogate models require many promising solutions [53]; ⊙ simulation data is often imbalanced & rough in unexpressed or noisy cases [36]; ⊙ approximation errors are increased accordingly, which may mislead search & selection [36].
S5 – $\Lambda$ : problem domain	◊ MMOPs (multitask) ◊ SMOPs (sequential) ◊ DMOPs (dynamic)	✓ Discriminator; ✓ Generator; ✓ Evaluator.	Transfer strategies: ⊕ without transfer by solving MOPs independently; ⊕ implicit transfer by assortative mating and selective imitation [54–56];	⊙ start from scratch when facing new coming MOPs, regardless of related optimization experience; ⊙ generalization abilities of traditional MOEAs are so poor that cannot leverage experience to achieve more promising optimization efficacy; ⊙ implicit knowledge transfer with random & blind transfer fails to improve overall performance.

The growing complexity of MOPs (e.g., MaOPs, LMOPs, and EMOPs) and real-world MOPs hardly appear in isolation have revealed the necessity of sharing optimization experience by processing a scaling-up number of  $K$  different MOPs in sequentially or simultaneously [37]. Accordingly, the resultant scaling-up problem is an SMOP or an MMOP with  $K \geq 2$ . Particularly, if we treat an MOP as a static state, then an SMOP will have multiple different states. When this sequential switch from one state to the new one can be simulated as a time-varying process, the resultant problem is a dynamic MOP (DMOP) [38]. Thus, a DMOP can be implicitly regarded as a sequential many-task MOP [39], in which all states are solved sequentially and the MOP in each state can be solved with the help of the knowledge or experiences accumulatively learned from previous optimization exercises [40].

### B. The Bottleneck of MOEAs for Scaling-up MOPs

Given an MOP to be solved, the appointed MOEA will iteratively run a stochastic optimization process on a population  $P$  to approximate the optima. As shown in Fig. 2, this process mainly includes three modules: 1) the generator for reproduction that reproduces a new offspring population  $Q$  by exploring the search space; 2) the function evaluator for calculating the objective values for each solution; 3) the discriminator for environmental selection that keeps the fittest solutions from the combined population  $P + Q$  to the

next generation. The potential challenges that circumscribe traditional MOEAs' generalization ability in handling scaling-up MOPs are summarized in Table I.

Traditional MOEAs face great challenges in responding to the ever-growing dimensionality and complexity of MOPs. These scaling-up MOPs spring up exuberantly in practical applications, e.g., in deep learning models with ever-expanding architecture and engineering systems with ever-complex linkage [21, 22]. Thus, to continue being applicable to realist, MOEAs should be equipped with learning ability to cope with new challenges brought by these scaling-up MOPs. The problem-specific features, if well learned with ML techniques, can be a great help to enhance the adaptive ability [57] and generalization ability [58] of MOEAs.

### C. Learnable Multiobjective Evolutionary Algorithms

In classic MOEAs, their generators and discriminators are developed based on some traditional genetic operators (e.g., crossover, mutation, and selection). These two evolutionary modules are unable to learn the exact characteristics of an MOP they encounter. Thus, they do not have the ability to respond flexibly to the potential challenges of solving this black-box MOP. Besides, most MOEAs are designed on the basis that all objective function expressions are known. However, an MOP may be unexpressed and noisy in practice. Without using additional measures to predict the exact expression in



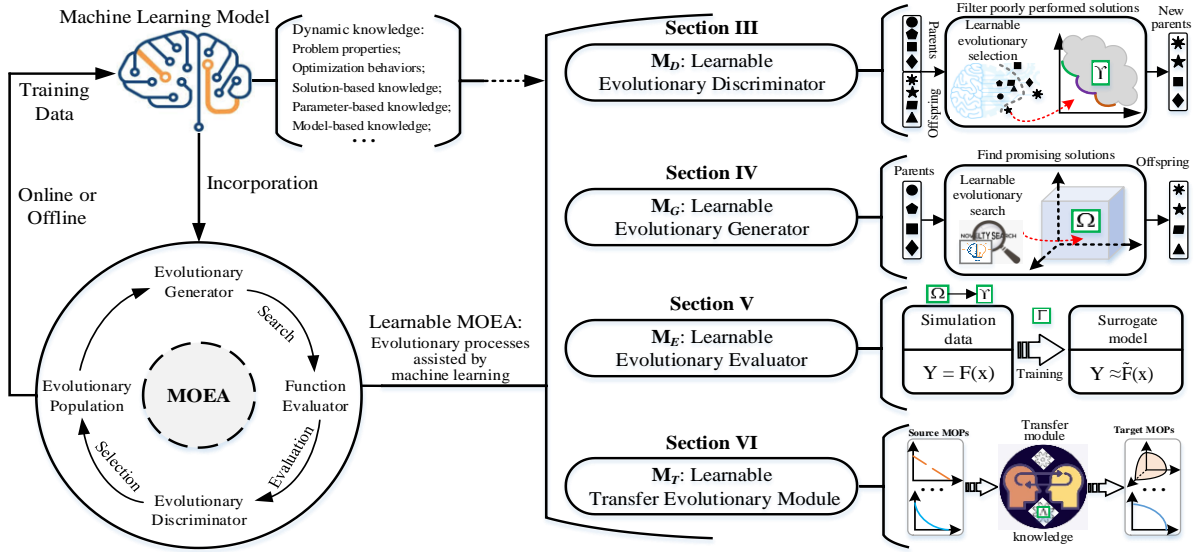


Fig. 3. Taxonomy of learnable MOEAs according to  $L_{MOEA}$  from its four learnable components, including  $M_D$ : evolutionary discriminator,  $M_G$ : evolutionary generator,  $M_E$ : evolutionary evaluator, and  $M_T$ : evolutionary transfer modules.

these uncertain cases, MOEAs are not able to work satisfactorily [36]. In addition, traditional MOEAs solve all MOPs independently, which is not an efficient manner when solving multiple MOPs sequentially or simultaneously. Many existing studies have shown that optimization efficiency can be significantly improved by transferring knowledge acquired from solving different (but somehow related) MOPs. In this context, learnable MOEAs are coupled with various ML techniques to assist their evolutionary processes so that they are capable of learning [59, 60]. Generally, learnable MOEAs build ML models to acquire potentially useful knowledge, e.g., problem properties, optimization behaviors, solution-based knowledge, parameter-based knowledge, model-based knowledge [40], etc. With proper guidance from the learned knowledge, they aim to improve the selection performance of its discriminator and the search capability of its generator [25]. When facing expensive MOPs, they expect to lightly predict the objective values of these MOPs using surrogate models, which are trained on the available simulation data collected from  $\Omega$  to  $\Upsilon$ . Moreover, it is sensible to create a transfer learning module between different MOP domains (often divided into source domains and target domains). The knowledge learned from solving source MOPs can be transferred to assist the optimization of target MOPs in a more efficient manner [29].

To clearly describe the “learning capability” of MOEAs, a struct  $L_{MOEA}$  with four learnable modules is defined below:

$$L_{MOEA} = (M_D, M_G, M_E, M_T) \quad (3)$$

where  $M_D$ ,  $M_G$ , and  $M_E$  are respectively the discriminator, generator, and evaluator of this MOEA;  $M_T$  indicates a learnable transfer module when this MOEA is used to solve scaling-up MOPs from the aspect of  $\Lambda$ . Thus, according to  $L_{MOEA}$ , existing learnable MOEAs can be roughly classified into four categories: MOEA with a learnable discriminator, MOEA with a learnable generator, MOEA with a learnable evaluator, and MOEA with a learnable transfer module, as

shown in Fig. 3. Their imperatives and the main scaling-up MOPs they are expected to solve are briefly described below.

**Learnable evolutionary discriminators:** ML techniques are used to subtly assist the environmental selection, which can handle the potential challenges brought by MaOPs with scaling-up objective spaces. With the increasing number of objectives, the ability to discriminate poor and elite solutions of existing environmental selection strategies deteriorates seriously. Specifically, the discriminator’s convergence pressure is insufficient and its diversity maintenance becomes unsatisfactory. Thus, learning how to improve the discriminative ability of selection strategies becomes particularly essential for solving MaOPs. The efforts made in this regard will be elaborately reviewed in Section III.

**Learnable evolutionary generators:** ML techniques are used to tactfully assist the evolutionary search, which can adaptively respond to the potential challenges brought by LMOPs with large-scale search spaces. The search capability of most existing genetic operators deteriorates dramatically in the large-scale search space, which leads to the fact that the ineffective offspring are often reproduced by the generator. Thus, learning how to enhance the search capability of the generators becomes extremely important for solving LMOPs. The efforts made in this regard will be comprehensively surveyed in Section IV.

**Learnable evolutionary evaluators:** Discriminators are customized based on solutions’ objective values, which are outputted by an evaluator. If the evaluator is blocked, e.g., in solving unexpressed MOPs, the discriminator will be unbootable. It is worth noting that the evaluator can be regarded as a discriminator to directly filter poorly performed solutions for single-objective optimization problems. In other words, the discriminator and evaluator are the same module in single-objective EAs, called fitness evaluator. To highlight the differences, this paper discusses them separately. The details of learnable evolutionary evaluators are provided in Section V.

**Learnable transfer evolutionary modules:** evolutionary modules via transfer learning are proposed to share useful optimization knowledge between source-target MOPs. Intuitively, it is a shortcut to solving the target MOP via reusing the experiences learned from the optimization exercises of solving source problems. The optimization of source MOPs can be completed or carried out at the same time as the target MOPs, and the respective resultant forms of transfer optimization are sequential form and multitasking form [40]. The target MOPs at hand are sequentially solved with the help of the experiences accumulatively learned from previous optimization exercises in the sequential form. By contrast, all the MOPs, starting from scratch, are optimized simultaneously in the multitasking form. The efforts made in learnable transfer evolutionary modules will be systematically investigated in Section VI.

### III. LEARNABLE EVOLUTIONARY DISCRIMINATORS

In an MOEA, its discriminator is customized to select the next population with balanceable convergence and diversity from the combined population of parent and offspring. As shown in Fig. 4, this section starts by answering what kind of dynamic knowledge the discriminator wants to learn in the objective space. Then, a review of existing efforts on learnable discriminators is given according to the types of ML methods used to acquire knowledge. Here, three traditional types of ML (i.e., supervised, unsupervised, and reinforcement learning) methods are mainly considered by the discriminator to improve its performance of environmental selection.

#### A. Learnable Discriminators via Supervised Learning

Supervised learning is defined as using labeled datasets to train a model or function that classifies future data or predicts outcomes accurately. With training data fed into the model, its parameters are dynamically adjusted until the model has been fitted appropriately. The dataset, used to train a supervised learning model, consists of the objective vectors of all candidate solutions in  $\Upsilon$ . Intuitively, a classifier model can be directly learned to predict the Pareto-rank of solutions in Pareto-based selections [61], to approximate the hypervolume contributions of solutions in indicator-based selections [62], and to evaluate the meta quality of solutions in decomposition-based selections [63]. However, these selection strategies of directly using a classifier model to filter solutions are often challenging when solving MaOPs with scaling-up objective spaces [64]. In this context, the relatively attractive way to assist the discriminator via supervised learning for handling MaOPs, especially for those with irregular PFs, is using an  $m - 1$  dimensional simplex model, which can dynamically approximate the geometry shape of their PFs.

In general, the geometry of the non-dominated front is estimated by training a regression model. Specifically, in some MOEAs [65, 66], the geometry of the non-dominated front can be predefined as the following simple curve:

$$(f'_1(x))^p + \dots + (f'_i(x))^p + \dots + (f'_m(x))^p = 1, p > 0 \quad (4)$$

where  $f'_i(x)$  indicates the  $i$ -th normalized objective value of a solution  $x$ , which is generally calculated as follows:

$$f'_i(x) = \frac{f'_i(x) - z_i^{\min}}{z_i^{\max} - z_i^{\min}} \quad (5)$$

where  $z_i^{\min}$  and  $z_i^{\max}$  are respectively the ideal and nadir points of the population in historical memory,  $i = 1, 2, \dots, m$ , and  $m$  represents the number of objectives. Obviously, this family of surfaces in (4) involves only one parameter (i.e.,  $p$ ) that needs to be learned, which reflects the curvature of the PF. Given a non-dominated solution set, termed  $NS$ , the value of  $p$  can be directly estimated by the least square fitting method as follows:

$$p = \arg \min_p \sum_{x \in NS} [(f'_1(x))^p + \dots + (f'_m(x))^p - 1]^2 \quad (6)$$

To reduce the computational cost, the prediction of  $p$  is limited to a predetermined and finite range of discrete values in some following works [67, 68]. Moreover, in [69], the PF of an MOP is assumed as the following linear model:

$$a_1 (f'_1(x))^{p_1} + \dots + a_i (f'_i(x))^{p_i} + \dots + a_m (f'_m(x))^{p_m} = 1 \quad (7)$$

where the hyperparameters are learned by maximizing the log marginal likelihood of Gaussian process regression in [70] and by the Levenberg-Marquardt algorithm in [71].

In this way, the learned PF's geometric properties (e.g., curvature, concavity, and convexity) can be used to guide the environmental selection for solving irregular MaOPs. Specifically, weights can be sampled adaptively [72] and scalarizing methods can be tailored properly according to the learned curvature of the PFs [73]. Thus, both the selection pressure and the maintenance of population diversity can be improved in decomposition-based strategies, as studied in learnable decomposition [70] and fuzzy decomposition [74] of MaOPs. According to the predicted model of the PF, adaptive performance indicators can be defined to properly evaluate the quality (convergence or diversity) of solutions in indicator-based selections, e.g., generic front model-based performance criteria [71], Minkowski distance [75], and direction similarity between solutions [74]. Furthermore, a Pareto-adaptive dominance (e.g.,  $pae$ -dominance [76]) can be defined and Pareto regions can be learned [77] to improve the selection pressure of Pareto-based strategies with the help of predicted geometries.

These simplex model-assisted selection strategies have been verified to be effective in solving MaOPs with irregular continuous PFs [78]. However, it is still a challenging task to maintain population diversity on MaOPs with disconnected, degenerated, and inverted PFs. To address this issue, some unsupervised learning-assisted selection strategies are proposed to enhance the performance consistency of MOEAs for tackling MaOPs with various PF shapes. The details are reviewed as follows.

#### B. Learnable Discriminators via Unsupervised Learning

Unsupervised learning uses ML algorithms to analyze and cluster unlabeled datasets. These algorithms discover hidden patterns or data distributions without the need of human

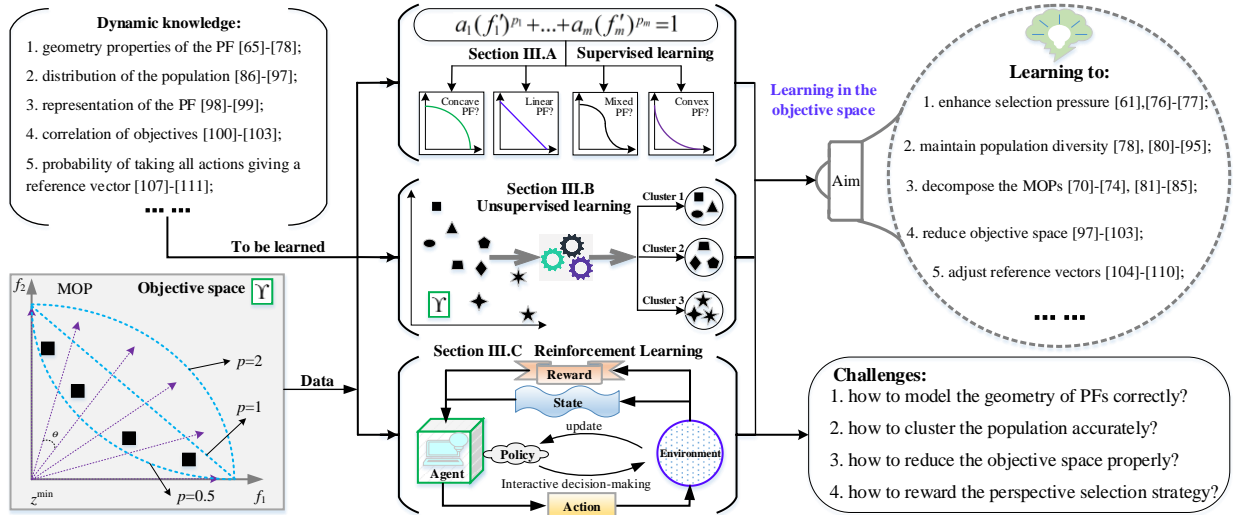


Fig. 4. Illustrating what knowledge can be learned in the objective space via various ML models and the aims of learning this dynamic knowledge.

intervention. Intuitively, clustering methods can be used to divide the population into multiple clusters in the objective space. The distribution information of solutions can be preserved through these clusters. The clustering results can assist the discriminators to select a population with balanceable convergence and diversity when handling MaOPs [79]. In the early Pareto-based strategies [80], the clustering methods with a Euclidean metric are used to prune the population when the size of the nondominated set exceeds preset bounds, which can make up the population diversity. However, these clustering-based pruning operators show insufficient selection pressure on scaling-up MaOPs as almost all solutions are mutually non-dominated from the beginning of evolution. In this context, decomposition-based strategies seem to stand out in terms of selection pressures for solving MaOPs. Various decomposition-based strategies are developed to divide the combined population into multiple mutually exclusive clusters according to a set of predefined reference vectors. Here, each solution belongs only to the cluster specified by its nearest reference vector [81]. Then, some representative solutions with better convergence are selected from each cluster. To be specific, only the solutions belonging to the same cluster are compared based on the adopted convergence indicators, e.g., locally weighted sum [82],  $\theta$ -dominance [83], angle-penalized distance [84], and clustering-ranking [85]. This constrained decomposition can be analogous to partition-based clustering with a direction similarity metric (e.g., the distance between two direction vectors), in which reference vectors are regarded as fixed centroids [81].

The main limitation of these fixed clustering-assisted selections is that their performance is strongly dependent on the matching between preset reference vectors and the target PF [46]. To circumvent this limitation, some up-and-coming strategies divide the population into multiple clusters via iteratively dynamic clustering methods without using reference vectors. For example, hierarchical clustering is used in [86–90],  $k$ -means and fuzzy  $c$ -means are adopted in [91], the Gaussian mixture model is applied in [92], density-based

clustering is employed in [93], and cascade clustering is used in [94]. These dynamically obtained clusters can match the geometric properties of the target PF adaptively. Concretely, the population is dynamically partitioned into multiple non-empty clusters for preserving population diversity, where solutions with higher similarity are gathered into the same cluster. Then, the representative solution of each cluster is survived into the next generation to maintain population convergence.

Nevertheless, dynamic clustering methods need to iteratively run many times to obtain final clusters with more balanceable diversity and convergence [95]. So there are three key concerns to be noted in designing dynamic clustering selections: 1) how to select a suitable clustering method? Although hierarchical clustering is commonly used to assist environmental selection, few efforts explain why it is the most appropriate. 2) How to select the similarity metric? Using Euclidean distance for clustering may lead to dominance-resistant solutions [96] being divided into a separate cluster. They have a high probability of surviving to the next generation, which will prevent the population from moving towards PF. To alleviate this issue, cosine similarity is widely used for clustering the population. The fairness of cosine similarity defined based on the ideal point is still dependent on the curvature of the target PF. It is often necessary to predict the curvature of PF in advance and then define the corresponding cosine similarity [95]. 3) How to select representative solutions in each cluster? The solution with the best convergence is always chosen as the representative of each cluster. Thus, it becomes critical to properly evaluate the convergence of solutions clustered in a locally constrained space.

In addition, reference vectors can be extracted adaptively by clustering to improve the performance of decomposition-based selection strategies [81]. The dimensionality of objective space can be reduced by clustering according to the conflict between objectives [97]. Intuitively, learning to reduce the number of objectives would yield shorter running times of selection, especially for hypervolume-based selection [98]. Meanwhile, an MaOP can be transferred to an MOP by eliminating

objectives that are not essential to describe the PF, followed by solving it with a typical MOEA [99]. Inspired by this, various dimensionality reduction methods, like principal component analysis (PCA) [100], semidefinite embedding [101], and transfer matrix with kriging model [102], are applied to reduce the objective space of MaOPs. Although the reduced representation of an MaOP will have a favorable impact on the selection efficiency, there are still three core concerns worth discussing and analyzing for objective space reduction [103]:

- 1) Whether it is possible to ignore some objectives while describing the PF of the target MaOP? and under what conditions are such objective reductions feasible?
- 2) How to define the correlation and conflict among objectives reasonably and quantitatively? and find out the insignificant or redundant objectives accordingly;
- 3) How to compute a minimum set of objectives according to such qualitative objective space changes?

Furthermore, adaptive reference adjustment via other unsupervised learning methods, e.g., self-organizing map [104] and growing neural gas [105, 106], is also a considerable way to promote decomposition-based selections in solving MaOPs.

### C. Learnable Discriminators via Reinforcement Learning

Reinforcement learning (RL) aims to learn the probability of taking an action, which will maximize the expected cumulative reward at the current state. Concretely, the agent takes an action at the current state based on the output of the policy network. Then, a reward is obtained after sending the action to the environment, followed by switching to the next cycle of inputting the new state and reward into the agent. This process will repeat until a terminal condition is fulfilled, as shown in Fig. 4. To solve a combinational MOP, the RL technique is applied in [107]. To be specific, an MOP is first divided into multiple subproblems with a set of reference vectors in the objective space, and then each sub-problem is modeled as a neural network. Subsequently, the model parameters of all neural networks are optimized by a deep RL algorithm. The optimal solutions can be directly obtained via these optimized models.

The key to improving the performance of decomposition-based discriminators is to redress the mismatch between the reference vectors and the PF shape. Given a finite number of actions, the most appropriate action can be selected to adjust the reference vector according to its current state and the probability of performing each action. This process can be realized by RL techniques. Inspired by this, an adaptive reference vector approach via RL is proposed in [108], in which the action space is defined as [*regeneration*, *relocation*, *return-back*, *state-keeping*] for the agent to control its reference vector. Besides, each agent of the reference vector can receive five different states according to the quantity of its associated dominated and non-dominated solutions. Finally, the Q-learning method [109] is used to learn the optimal policy network. An RL-assisted decomposition-based discriminator is also developed in [110]. To the best of our knowledge, there are few efforts to improve the performance of environmental selection via RL, but this does not mean it is not a promising

research direction. For example, RL techniques have been used to automatically design an appropriate discriminator and set optimal parameters when facing different MaOPs [111].

## IV. LEARNABLE EVOLUTIONARY GENERATORS

With the inputted parent solutions, an evolutionary generator aims to reproduce offspring solutions that can move in the promising direction to approach the unknown PS fast. To achieve this, it explores the search space by processing parent solutions via genetic operators. However, the randomness-based genetic operators are often unable to grasp the direction and pace of the search, which may drive the population to move forward roughly and slowly. Consequently, greatly high amounts of computing resources are required to approach the PS when solving LMOPs. As computing budgets are often limited in practice, it becomes particularly essential for a generator to simplify the large-scale search space or guide the population to converge fast along with the promising directions (e.g., improve its search capability). In this context, ML techniques are widely used to discover the structural information of search space and the latent promising converging paths of solutions excavated from available data [112]. The learned model can then be applied to find ways of guiding evolutionary search with high efficiency in responding to the challenges brought by the scaling-up LMOPs.

This section will start with a brief review of learnable generators in multiobjective estimation of distribution algorithms (MEDAs), which directly learn a probabilistic model to sample solutions. The limitations of MEDAs in solving scaling-up LMOPs are discussed thereafter. Then, the following survey route is outlined in Fig. 5. Similarly, this section focuses on three main types of ML (i.e., supervised, unsupervised, and RL) methods and investigates how they can be used to enhance the search capability of genetic operators.

### A. Learnable Generators via Estimation of Distribution

To mitigate the adverse influence (e.g., the disruption of building blocks and mating restrictions) led by genetic operators, the evolutionary generator is directly designed based on a probabilistic model built to approximate the PS in most MEDAs [113]. Here, the probabilistic model can be a Gaussian distribution [114], a Bayesian network [115], a regression decision tree [116], a restricted Boltzmann machine [117], a growing neural gas [118], a manifold learning model [119], etc. In this way, new solutions can be directly sampled with the probabilistic model. It is progressively learned from the statistic information (e.g., the dependencies between variables and the regularity property of the PS), which is captured from the existing solutions. Clearly, the model-building is the key step of MEDAs, which leads to the following four main classes of MEDAs based on the adopted modeling strategies:

- 1) Graphical model-based MEDAs learn a Bayesian network (a directed acyclic graphical model, in which the nodes represent the variables and the arcs indicate the dependencies) as the probabilistic model for solution sampling [120, 121].





consuming functional evaluation compared to traditional local search methods. In extensions of *innovization*, the concept of knowledge-driven optimization is introduced in [144], in which MOEAs absorb the knowledge (e.g., the latent patterns shared by high-quality solutions) learned from intermediate solutions to guide their search away from mediocre solutions and toward potentially promising areas. This is an online learning process from deciphering what makes one solution optimal (or near optimal) in the final obtained solution set and revealing what makes one solution dominated by (or perform better than) another during the optimization process. Online *innovization* aims to accelerate the evolutionary search and thus improve the efficiency of generators [145].

In an online *innovization* operator, generally, a supervised learning model is first built and trained online. It expects to implicitly learn the historically directional improvements (e.g., from dominated to non-dominated) of solutions in the already explored search space. As shown in Fig. 5, a solution  $x = (x_1, x_2, \dots, x_n)$  is identified to be performing poorly, while  $\vec{x}$  is of high quality. During the evolutionary process, a variety of  $(x, \vec{x})$  solution pairs can be collected. Then, the adopted model, e.g., a multilayer perceptron [146] a random forest [147], or a deep neural network [148, 149], can be trained using these labeled data. Here,  $x$  is the input and  $\vec{x}$  is its label, i.e., the expected target output. In this way, the trained model is believed to be able to capture the underlying pattern that reflects the directional improvement of solutions in the search space. Expectantly, a new offspring solution  $x^{\text{new}}$  generated by genetic operators can be repaired (or progressed) via inputting it into the well-trained model to get its improved version  $y^{\text{new}}$ . This process of repairing offspring is called *innovized* progress. It's possible to improve the search capability of generators in solving scaling-up MOPs due to its following four merits: 1) involving all conflicting objective information; 2) without requiring the cost of function evaluations; 3) the learned directional improvement of solutions is not fixed but is adaptive with the elapse of generations; 4) the moving from  $x^{\text{new}}$  to  $y^{\text{new}}$  by the learned model in the search space may result in a quantum leap in the objective space [147]. Nonetheless, there are also four key considerations when customizing such an *innovization* progress:

**Selection of the learning model:** It's flexible to choose potentially available supervised learning models to match the requirements. But, it's also reasonable to consider the actual computational cost caused by learning the selected model. For example, the random forest and multi-layer perceptron are respectively used in [147] and [146] owing to their lightweight computational efficiency when compared with the deep neural networks adopted in [148, 149].

**Collection of training data:** Paired training data, either from the previous or the current generation, are collected based on the performance of available solutions. Thus, when deciding to collect a pair  $(x, \vec{x})$ , it should consider that  $\vec{x}$  performs better than  $x$  (e.g., dominates  $x$  in a Pareto-based MOEA [141], the aggregation fitness of  $\vec{x}$  is better than  $x$  in a decomposition-based MOEA [149], and  $\vec{x}$  is most likely to guide  $x$  to converge rapidly [146]). Besides, the labeled population consisting of all  $\vec{x}$  solutions should perform better

than the input population with all  $x$  solutions in terms of convergence-diversity tradeoff.

**Training of the adopted model:** Training a model itself is also an optimization problem, which involves many hyperparameters (e.g., model architecture, learning rate, and training epoch) that need to be set manually. Thus, many other issues may be encountered (e.g., overfitting). It is also worth exploring whether the model needs to be updated every generation (e.g., online or offline training).

**Advancement of the search capability with the learned model:** The expectation is that the search capability of the generator can be improved with the aid of this well-trained model. Specifically, the poor and mediocre solutions in the population can be repaired to converge fast in the learned promising direction. Meanwhile, high-quality solutions can be advanced to seek more diverse elite solutions. Nevertheless, it involves the following two considerations: 1) is it necessary to repair all the newly generated solutions? 2) is it necessary to run the *innovization* progress in every generation?

Given above, the supervised learning methods are mainly used to improve the search capability of the evolutionary generator without simplifying the original search space.

### C. Learnable Generators via Unsupervised Learning

As discussed in Section IV.A, the population is first divided into multiple clusters in some MEDAs by a clustering method. Then a probabilistic model is created on each cluster so that the created mixed model has a strong generalization ability in sampling solutions. Similarly, various clustering methods, e.g.,  $k$ -means [150], spectral clustering [151], manifold clustering [152], and  $k$ -nearest neighbor (KNN) based graph [153], are applied to assist evolutionary generators. To be specific, the crossover in a generator is restrictedly run only among solutions belonging to the same cluster. A plausible belief behind this is that mating similar solutions can create new solutions with higher quality and thus can improve search capability. Besides, different genetic operators can be implemented in different clusters instead of using the same search operators for the whole population, which can balance the exploration and exploitation of evolutionary search.

Additionally, clustering methods are used in [154, 155] to classify the decision variables into multiple groups based on their contributions to the objectives. To be specific, all variables are divided into convergence-related and diversity-related groups based on the data collected by a variable contribution analysis. Afterward, different genetic operators are designed to search the subspace respectively consisting of two groups of variables, which can adaptively balance the convergence and diversity. Here, the clustering of variables requires analyzing the contribution of each variable (partial to convergence or diversity). Thus, this variable analysis process will consume a lot of function evaluations. Consequently, this kind of method often fails to do actual optimization of LMOPs when allocating a limited budget [140]. Similarly, the decision variables are classified into highly and weakly robustness-related groups in [156] based on their contributions to the robustness of candidate solutions. Nonetheless, it is not

enough to divide all variables into two groups because there are still many variables in each group for LMOPs. Therefore, the dimension of each group is further reduced by PCA in [157]. In this way, a low-dimensional representative space can be obtained, and the generator can get a fast convergence speed by searching in this small-scale space.

The key question here is whether there is such a representative space for an LMOP. In this context, the concept of sparse LMOP arises [158], in which the values of most variables in optimal solutions are 0. The PS of a sparse LMOP can be approximated by finding a Pareto representation in a latent small-scale representative space. Since sparse LMOPs widely exist in real-world applications [159] (e.g., neural network training, neural architecture search, community detection, signal processing, portfolio optimization, etc.), a variety of dimensionality reduction techniques (such as unsupervised neural networks [160], data mining [161], logistic PCA [162], and network embedding [163]) are applied to learn the representative space (also termed Pareto subspace) of sparse LMOPs. After that, new offspring can be generated by searching in this learned subspace, including the following three steps:

- 1) **Encoding:** the solutions of an LMOP are encoded as small-scale representations;
- 2) **Searching:** the genetic operators are run on encoded representations to generate new representations;
- 3) **Decoding:** the newborn representations are mapped back to the original search space.

Consequently, the strategy of representation learning is proved to be effective in solving LMOPs with high sparsity.

Another promising way to improve the search capability is to adaptively select genetic operators or adaptively control/tune their parameters. Existing search strategies often need to set various hyper-parameters and exhibit quite different preferences in search patterns and scopes. Learning from historical search experience, some recent studies have been dedicated to the adaptive selection of better operators or the adaptive setting of appropriate parameters [164, 165]. For example, the AdaBoost-inspired multi-operator ensemble is proposed in [164] to define the credit assignment scheme for adaptively rewarding the involved operators. The clustering-based fuzzy system is customized in [165] to adaptively tune the crossover and mutation probabilities. The key component here is the credit assignment scheme used to assess the performance improvement brought by each participant operator [166]. Then, the higher-performing operator is activated by the generator to reproduce offspring in the next generation.

#### D. Learnable Generators via Reinforcement Learning

Adaptive operator selection or adaptive parameter setting are controllers in improving the search capability of evolutionary generators [167]. However, relying only on historical rewards, they still suffer from the dilemma of exploration versus exploitation. Accountably, the operators may be inaccurately punished due to the fact that one who performs poorly in previous generations may be effective in future generations. Besides, searching with poorly assessed operators according to

their historical rewards may be beneficial for the solutions to escape from the local optimum [168]. To alleviate these issues, RL-assisted control strategies are proposed in [169–174] and [175–178], respectively, for operator selection and parameter setting. Compared with the adaptive control strategies, they have the following advantages:

- 1) RL-assisted strategies rely on future rewards rather than historical rewards, which expect the selected operators to be effective in future generations rather than in the previous generations;
- 2) RL-assisted strategies can provide a more delicate decision on operator selection as they consider the characteristics of each solution. Besides, they can assign different operators for different parents rather than reward operators equally for all solutions.

Since the future rewards are unknown during the search process, it is needed to be deduced by training a relatively complex RL model. It includes the following four core components to be determined:

**Actions:** a set of genetic operators constitutes the discrete action space. Here, the same operator with different parameter settings can be considered as a different operator.

**States:** the decision variables of solutions defined in the search space (i.e., the environment) are regarded as states, meaning that each solution has its own state.

**Policy network:** Inputting a state, a policy network should output the probability of taking each action so that the agent can decide a reasonable next action. Generally, the probability of a decision increases for high reward values. To train the policy network steadily, the reward for a state-action pair should be set reasonably. It should take both convergence and diversity into consideration in the performance improvement obtained by newly generated offspring.

**Rewards:** Since the action space is discrete, the classical Q-learning can be used to approximate the action-value function [168], which is always defined based on the discount sum of future rewards. Here, the reward is calculated based on the performance improvement of offspring obtained after taking a genetic operator. The reward corresponding to these actions is stored in a Q-table, where each grid represents the expected cumulative reward of an action given a state.

In short, the agent selects a long-sighted operator (i.e., action) to the MOEA according to the output of the policy network. Then, an offspring (i.e., a new state) is generated by using the selected operator to search in the variable space (i.e., environment). After that, the new state and performance improvement (i.e., reward) are sent to the agent for updating the policy network. With existing updated rules for Q-learning in place, the core of designing RL-assisted control strategies shifts to assessing the performance improvement from parents to offspring brought by the conducted genetic operator. Thus, the discriminators introduced in Section III can be used here to get a reasonable assessment of the performance improvement.

## V. LEARNABLE EVOLUTIONARY EVALUATORS

For MOPs with clear expression, the MOEA can directly use its evaluator to calculate the objective vector for each

solution  $x$ , i.e., the given function vector  $F(x)$ . However, the objectives of many practical MOPs often lack a closed form and may be expensive to evaluate from simulation data, which makes MOEAs unable to get the near-optimal solution set within a limited computing time [179]. To address this issue, learnable evaluators use surrogate models to simulate the objective functions or fitness of solutions. Correspondingly, learnable evaluators can be roughly divided into two different types: learnable function evaluators [180] and learnable fitness evaluators [181]. The details of them are reviewed below.

#### A. Learnable Function Evaluators

For learnable function evaluators, the calculations of real expensive objective functions are simulated by directly training a cheap regression-based surrogate model, e.g., Kriging model [182–184], dropout neural network [185], radial basis function [186, 187], hybrid or ensemble models [188, 189]. In this way, to save computing time in each generation during evolution, only a small number of promising solutions (sampled by the surrogates) are eligible for real function evaluations. Then, they are used to update the surrogates, which are applied to evaluate the remaining solutions functionally. Thus, the trained inexpensive surrogates often need to be accompanied by the model management to sample promising solutions. Here, various infill sampling criteria are proposed for model management [190], such as lower confidence bound, probability improvement, expected improvement, and multiobjective infill criterion [191]. Several survey papers have been proposed to detail surrogate-assisted EAs. Interested readers should refer to these related reviews [36, 179, 192].

#### B. Learnable Fitness Evaluators

In addition to directly simulating real function evaluations, some learnable fitness evaluators switch to building surrogates for approximating scalar aggregation fitness of multiple objectives in decomposition-based MOEAs [193, 194] or for simulating a performance indicator (like hypervolume) in indicator-based MOEAs [195, 196]. Besides, the evaluation of objective functions may be less important in solving an MOP if the dominant relationship between solutions is known. Inspired by this, some learnable fitness evaluators expect to skip the function evaluator, and directly learn a classifier to determine the quality or dominance levels of solutions [197]. Ideally, it is possible to learn a binary classifier to determine which solutions should survive into the next generation (positive class) and which should not (negative class) [198]. Thus, these classification-based evaluators, also acting as discriminators, can concurrently balance the convergence and diversity of population if an appropriate classification boundary is defined [199]. To learn Pareto-dominance relationship between solutions, ten off-the-shelf classification models are studied in [200]. Deep feedforward neural networks are used to predict the  $\theta$ -dominance levels of candidate solutions in [201] and to learn Pareto-dominance relations between solutions and references in [202]. KNN model is utilized in [203] for filtering candidate solutions and the support vector machine is employed in [204, 205] to classify solutions based on each subproblem's scalarization fitness.

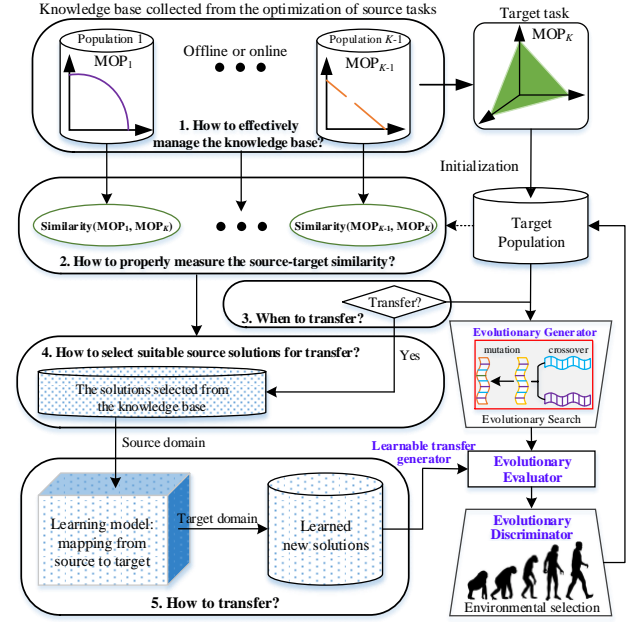


Fig. 6. The basic framework of solution-based transfer generators and the main challenges when designing this kind of transfer generators.

## VI. LEARNABLE TRANSFER EVOLUTIONARY MODULES

Taking the evolutionary process of solving an MOP as an optimization exercise (or task), the aforementioned MOEAs generally show the following three characteristics: 1) they often start an evolutionary search from scratch with a random population; 2) they assume that all MOPs are mutually independent and thus solve each MOP as a separate task; 3) their problem-solving ability does not grow with the optimization exercises they have undergone. However, such an optimization manner of solving each MOP independently is inconsistent with the problem-solving behaviors of human beings [206]. A prospective MOEA should be able to improve its problem-solving ability by accumulating experience from previous optimization exercises. Then, it can solve relevant MOPs at hand in a more efficient way. Taking this cue, research into reusing transferable knowledge (from the experiences of optimizing source MOPs) to assist in the optimization of target MOPs has attracted extensive attention. This kind of learnable MOEAs is always coupled with transfer learning [29]. They expect to find such a shortcut that the evolutionary population could 1) have a higher starting state than starting from scratch; 2) have a faster convergence rate; and 3) eventually be closer to the target PF or PS. This section will sort out the efforts in improving the performance of three evolutionary modules (i.e., generators, discriminators, and evaluators) through transfer learning. The details are provided and discussed below.

#### A. Learnable Transfer Generators

A learnable transfer generator continually updates and enriches its search experience by learning from the optimization of source MOPs [40]. Then, its search efficiency can be improved with the assistance of transferred knowledge. Existing efforts on learnable transfer generators mainly regard solutions

of source MOPs as knowledge to be transferred. As shown in Fig. 6, solution-based transfer generators are primarily studied by addressing the five key challenges, as follows:

**How to effectively manage the knowledge base?** The discovered optimal solution sets in solving source MOPs are collected for updating the knowledge base. Here, a threshold is always preset to determine the number of saved solutions for each source MOP. Once the cardinality of a source MOP in the knowledge base exceeds the threshold, some related solutions need to be filtered out. For example, the source task completed later is believed to be more closely related to the target task in solving a dynamic MOP [207]. Thus, the optimal populations of adjacent source tasks are preferred to be saved when pruning the knowledge base [208]. For periodic dynamic MOPs, the source tasks at the same cycle stage as the target task are preferred by the knowledge base [209]. It is noteworthy that dynamic optimization is regarded as a special type of sequential optimization as shown in Fig. 2. For more information on dynamic optimization, please refer to this latest survey [210]. When solving multiple MOPs simultaneously from scratch, the optimization of each MOP is both a target task and a source task [40], in which the knowledge base always consists of the current populations they explored.

**How to properly measure the source-target similarity?** The negative transfer may occur when the PS of the source MOP is unrelated or even opposite to that of the target MOP. To alleviate this issue, it is essential to exactly measure the source-target similarity based on the distribution distance between their obtained solutions. Various measures of distributional distances, e.g., mutual information [211], Wasserstein distance [212], Kullback-Leibler divergence [213], and maximum mean discrepancy [214], have been used to quantify the similarity between source-target MOPs. Since the obtained solutions change with the evolutionary process, the source-target similarity should be always updated online, which is often intractable. For example, change detection in dynamic optimization is implicitly a process to measure whether two adjacent tasks are similar or not. Most of the related works learn a prediction model to detect environmental change [215]. However, the solution sets collected in different environments for prediction do not always follow the same distribution [216], which will mislead the search. In heterogeneous source-target scenarios, it is often necessary to do padding or map all the solutions to a common space [213]. In multitasking scenarios, the online population may only cover a subregion of the search space, which further increases the difficulty of online similarity measurement. Thus, the inter-task gene similarity is defined in [217] by a probabilistic model to feature each gene.

**When to transfer?** Too frequent knowledge transfer needs to spend a huge amount of function evaluation on the transferred solutions, which is not conducive to efficient optimization [218]. It will also lead to a large increase in the probability of negative transfer. Besides, solutions transferred to the target MOP may be only useful at certain evolutionary stages. Thereby, it is worth discussing whether it is necessary to transfer knowledge from source and when is appropriate to transfer. For example, knowledge transfer is generally put forward only when the environment change is detected in

dynamic optimization [219].

#### **How to select suitable source solutions for transfer?**

The types of source tasks will be diverse with the increase in optimization experiences. Accordingly, the selection of appropriate source tasks for transfer learning becomes critical. The most intuitive way is to select the source task with the highest similarity to the target task. However, due to the potential randomness of evolutionary search, the population's distribution is constantly changing during the search process. Thus, the current source-target similarity hardly reflects their actual relatedness. In fact, without a *priori* knowledge about source and target MOPs, it is difficult to correctly judge the difference between source and target from their obtained solutions [220]. Nevertheless, the source-target similarity suggests some latent connection between them. A variety of strategies for selecting source tasks based on the source-target similarity are presented in [212], including the nearest selection, weighted selection, top- $k$  selection, and their mixed version. In [221], according to the contribution of variables, the source tasks saved in the knowledge base are divided into diversity-related and convergence-related source tasks. In [222], multiple highly similar source tasks are first selected for each target task, which will learn from these multiple sources using a local distribution estimation.

In addition, all the solutions of source tasks are potential transferable knowledge, some of which are helpful while others may be futile or even counter-productive. Hence, the selection of solutions with high-quality knowledge is also an important foundation to underpin positive transfer. In [223], each solution will get a class label ("+" or "-") via an incrementally trained naive Bayes classifier, and then only the solutions whose predictive class labels are "+" can be the candidate transferable solutions. In [224], the transferred solution will be marked as positive once it is non-dominated for the target task, and then only neighbors of positive transferred solutions can be the candidates for future knowledge transfer. The concept of "transfer rank" is defined in [225] to further quantify the priority of each historical positive transferred solution, in which those solutions with higher ranks are preferred to be transferred. In [226], a knowledge estimation metric is defined to imply the effectiveness of solutions. In [227], each subproblem of the target task maintains an external neighborhood with the subproblems of source tasks via grey relation analysis. Then, the neighborhood information acts as a bridge to limit the selection of transferable solutions.

**How to transfer?** In homogeneous source-target scenarios, the solutions from the source domain can be directly regarded as a new solution in the target domain, such as in most dynamic optimization cases [210]. However, when facing heterogeneous source-target scenarios, a suitable transfer model is required to be learned for mapping solutions from the source domain to the target domain. For example, a source solution  $y$  is mapped into the target domain by an autoencoder model, which can generate a new solution  $x$  with the linear mapping  $f: x = yM$  [228–230]. Here, the matrix  $M = (QP^T)(PP^T)^{-1}$  is learned by minimizing  $\|QM - P\|^2$ , where  $Q$  and  $P$  respectively represent the populations for source and target MOPs. For achieving nonlinear mapping,



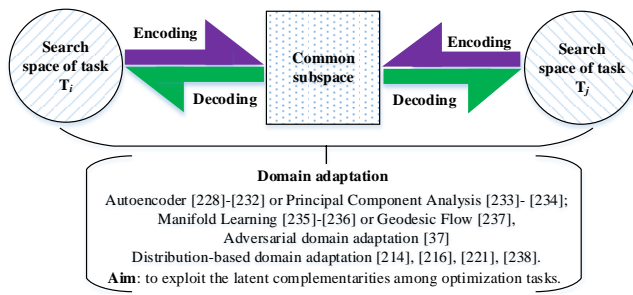


Fig. 7. The basic process and basic idea of a domain adaptation based on common subspace learning.

the kernelized autoencoder is developed in [231] and [232] for solving sequential MOPs and dynamic MOPs, respectively.

Moreover, a robust transfer model should be built by considering the distribution shift between source and target tasks. Domain adaptation methods are designed to overcome this shift, which can reduce the discrepancy between source and target tasks by learning their domain-invariant feature representations. Specifically, an intermediate subspace is constructed to find a learnable alignment matrix that transforms a source solution into the target one. As listed in Fig. 7, the subspace can be composed of several most significant eigenvectors (representing the necessary features) induced by PCA, as studied in [233] and [234]. A very similar method that treats the search space as a manifold is proposed in [235, 236]. Then, all search spaces modeled as a joint manifold are projected into a latent subspace by solving a generalized eigenvalue decomposition problem. The coordinate system is defined in [237] to identify some common modality in a proper intermediate subspace, which is generated along the geodesic flow starting from the source subspace to the target subspace. Furthermore, domain adaptation based on probability distribution is also widely used to sample transferable solutions, e.g., the transfer component analysis used in [214], the joint distribution adaptation studied in [216], and the moment distribution matching and representation learning of source-target are respectively proposed in [221] and [238].

Most efforts on learnable transfer generators focus on reusing solution-based knowledge. However, in practice, the representations of knowledge should be diverse, such as the trend of environmental change in dynamic optimization, the search behaviors of different genetic operators, and the latent attributes that affect evolution [239]. Thus, the search trend is learned in [240] by a maximum likelihood estimation model, adaptively controlling the evolutionary search (e.g., search step size and range). A neural network and a SVM are respectively trained in [241] and [242] to learn the transfer trend of environment changes for reproducing promising solutions in new environments. In [243], the concept of "meta-knowledge" is introduced to evolve the task-specific knowledge.

### B. Learnable Transfer Evaluators and Discriminators

Existing efforts on evolutionary transfer optimization mainly focus on improving the search capability of generators via solution-based knowledge transfer. There are still few efforts

on learnable transfer evaluators and discriminators for EMO. This subsection will summarize the related studies. Among the basic challenges, how to alleviate negative transfer and ensure maximum positive transfer, even in scenarios of source-target domain mismatch, is also the focus of this research area [244].

Learnable transfer evaluators are mainly designed to handle EMOPs. Multiple surrogates are built respectively for different computationally expensive tasks, which are jointly solved to reduce the number of function evaluations [245]. Intuitively, transferable knowledge embedded in cheap surrogates can be reused to avoid computationally expensive surrogate modeling [246]. For example, the evaluation of different objectives may involve distinct computer simulations with different computational complexities. To evaluate these non-uniform objectives, different surrogates should be created. In this scenario, the useful knowledge (e.g., parameters) from the surrogates for cheap (or fast) objectives can be transferred to assist the surrogates for expensive (or slow) objectives [247, 248]. Besides, auxiliary (or local) surrogates trained on subsampled data (or local data) can be regarded as helpful knowledge to assist the learning of global surrogates on all available data (or large data) [249, 250]. For solving EMOPs in dynamic environments, the previously archived data is transferred to the new environment to jump-start (or warm start) its surrogate modeling process [251, 252]. In [253], the knowledge learned in distinct surrogate models is transferred to augment the optimization of new possible related target tasks. To avoid the complex covariance evaluation in Bayesian optimization, multitask conditional neural process nets are built as surrogate models to replace the Gaussian process. The proposed surrogate models augment the observed dataset with several related tasks to estimate model parameters confidently [254].

The evaluators discussed above can be regarded as a discriminator in certain cases, such as when there is only one objective function per task. In addition, a discriminator based on transfer learning between different clusters is proposed in [255], which aims to distinguish the quality of solutions in the new environment. Ensemble prediction models are learned in [256] to find an acceptable but more robust PF during environmental changes. An RL model is built in [257] to respond to the movement and different severity degrees of the Pareto front in time-varying environments. For EMOPs and DMOPs with large-scale search spaces, it may be a promising research direction by transferable inverse modeling [258–260]. In [261], a multi-step nonlinear regression is designed to quantify the convergence status and estimate the intensity of knowledge transfer, aiming at achieving adaptive allocation of computational resources in multitasking scenarios.

Finally, an MOEA can collect a large amount of diverse data and optimization knowledge after solving a series of different MOPs, but how to reuse such data and knowledge smartly in promoting the future optimization task is still in its infancy [262]. The intention of this section is not to review the existing work on multitasking, expensive, dynamic optimization, etc., but to give an investigation of knowledge transfer in these different optimization scenarios [263]. Interested researchers can refer to their latest surveys [20, 179, 210, 264] for getting insight into more details. Here, we suggest a recent short letter

[265], which summarizes the fundamental research questions, limitations, challenges, and directions for the future related to evolutionary transfer or multitask optimization.

## VII. CHALLENGES AND FUTURE OPPORTUNITIES

Despite the efforts of MOEAs powered by ML methods, the potential of learnable MOEAs for scaling-up MOPs has not yet been fully explored. These efforts are only the first steps and still have much room for improvement [266]. Here, we discuss the challenges in advancing learnable MOEAs and provide insights on the prospects of future opportunities.

### A. Challenges

At first, both MOEAs and ML methods are insufficiently studied in theory. The theoretically fundamental studies of learnable MOEAs are challenging. It may be difficult to mathematically define the diverse types of knowledge learned during evolution. It's also hard to quantify how much the learned knowledge can help MOEAs in solving various scaling-up MOPs. Furthermore, the theoretical convergence and computational complexity analysis of learnable MOEAs may be more difficult to advance [267].

Secondly, the understandability and robustness of learnable MOEAs for MOPs are challenging. MOEAs often generate changeable and locally distributed data, and ML is an optimization task of function approximation from given data. Thus, it may be challenging to learn the expected function (e.g., what we want the ML model to do). Consequently, the approximated function of the adopted ML model may be noisy and even contain contradictions to the target optimization of MOPs. Therefore, there is still a long way to better understand exactly what ML models can contribute to evolutionary optimization tasks and to clearly explain why the adopted ML model works well on specific optimization tasks.

Thirdly, developing efficient learnable MOEAs for various scaling-up MOPs remains an open challenge. ML models are usually susceptible and computationally expensive. There are so many existing ML models and they are still being updated rapidly. To efficiently solve the scaling-up MOPs of interest, it is difficult to determine what kind of ML models should be assembled into MOEAs. It is rarely wise that research studies provide a tiny accuracy benefit at the cost of massive increases in model complexity. Even the addition of one or two seemingly innocuous data dependencies can slow down the progress.

Fourthly, great difficulties remain regarding the fairness and verifiability of performance comparison for newly developed learnable MOEAs. As explained in [268], it is difficult to guarantee absolute fairness in performance comparison of MOEAs, which involves various factors to be considered. Specifically, different population sizes, test problems, performance indicators, or termination conditions may result in different performance comparison results. Besides, most of the existing ML models are inherently difficult to explain in terms of performance. Current studies verify and analyze the performance of a new learnable MOEA in an intuitive way, which is not beneficial for the development of research in this

field. Thus, it is of great significance to take this aspect into consideration to endow learnable MOEAs with scientific rigor and transparency [269].

Last but not the least, there is still a long way to go to apply learnable MOEAs for tackling real-world MOPs. The performance of most learnable MOEAs is validated only by synthetic MOPs. Whether the real challenge of solving these practical MOPs is the same as the synthetic MOPs is a very meaningful but challenging inquiry.

### B. Future Opportunities

In order to drive the development of learnable MOEAs for solving scaling-up MOPs, several potential directions and opportunities for future study are provided below.

One direction is to improve the effectiveness and efficiency of learnable MOEAs. With this goal, we should prefer to couple an ML model with high performance and low computational cost into MOEAs, rather than deliberately copying the popular ML models. The high performance here should depend on whether the model can exactly contribute to improving the problem-solving ability. The low computational cost should not just be about the function evaluation cost in training an ML model. Learning the model is also an optimization process, so determining how to achieve a better balance between learning and optimization is a meaningful topic for future study. For example, design lightweight classifiers to replace the regression-based function evaluations for expensive MOPs [270] and combine some local search methods to accelerate the population's convergence speed [271].

Designing learnable MOEAs to solve more complex or special MOPs is also a promising research direction, e.g., surrogate-based Q-learning for dynamic time-linkage EMO [272], surrogate models for high-dimensional expensive EMO [273], adaptive surrogates for expensive constrained multimodal EMO [274], evolutionary transfer for multi-scenario EMO [275], multitasking under interval uncertainties [276], evolutionary multitasking for constrained EMO [277, 278], federated learning for secure EMO [279]. Besides, we should not be satisfied with effectively solving synthetic MOPs, but pursue more smart or intelligent optimizers for practical applications, e.g., multitasking and hyper-heuristic are unified in [280] with a graph-based framework [281], hybridization of MOEA and RL for multiobjective orienteering optimization [282]. Thus, designing various scaling-up MOP benchmarks closely related to practical applications is another meaningful topic that can further advance this field [283], e.g., data-driven MOPs [284], benchmarking neural architecture search as MOPs [285], and benchmarking feature selection as MOPs [286].

Advances in hardware and software (e.g., GPU and parallel computing [287]), facilitating larger models and faster training, have greatly promoted the research of deep learning. Similar technological advances for learnable MOEAs could massively improve their problem-solving efficacy [288]. If possible, bespoke hardware and software could be developed to run learnable MOEAs for tackling the challenges of ever scaling-up MOPs we may encounter in the future. For example, the

GPU-based knowledge transfer mechanisms are proposed in [289] to handle many-task scenarios and the batch recommendation approach is presented in [290] to evaluate multiple samples parallelly for data-driven optimization [291]. Finally, the automated design of learnable MOEAs is also a promising development direction [292].

### VIII. CONCLUSIONS

When solving the challenging scaling-up MOPs, designing effective learnable MOEAs has become a hot topic in evolutionary multiobjective optimization due to their competitive performance and tremendous potential compared with traditional model-free MOEAs. Equipped with various machine learning techniques, different types of learnable MOEAs, as summarized in this survey, have been proposed in recent years. These learnable optimizers show promising problem-solving ability in responding to the grand challenges from a wide scale of complex MOP scenarios, e.g., MaOPs, LMOPs, EMOPs, DMOPs, SMOPs, MMOPs, etc. To clarify the underlying collaboration of applying ML methods to assist MOEAs, this paper defines the learnability of MOEA from four different learnable evolutionary modules, i.e., discriminator, generator, evaluator, and transfer modules. Then, a systematic review is provided for the advancement of each direction. Finally, the challenges of developing learnable MOEAs are provided and several future opportunities are presented to drive the development of learnable MOEAs for scaling-up MOPs.

### REFERENCES

- [1] Y. Jin and J. Branke, "Evolutionary optimization in uncertain environments—a survey," *IEEE Trans. Evol. Comput.*, vol. 9, no. 3, pp. 303–317, 2005.
- [2] E. Alba and M. Tomassini, "Parallelism and evolutionary algorithms," *IEEE transactions on evolutionary computation*, vol. 6, no. 5, pp. 443–462, 2002.
- [3] C. A. Coello Coello *et al.*, "Evolutionary multiobjective optimization: open research areas and some challenges lying ahead," *Complex & Intelligent Systems*, vol. 6, pp. 221–236, 2020.
- [4] A. Arias-Montano, C. A. C. Coello, and E. Mezura-Montes, "Multiobjective evolutionary algorithms in aeronautical and aerospace engineering," *IEEE transactions on evolutionary computation*, vol. 16, no. 5, pp. 662–694, 2012.
- [5] A. Ponsich, A. L. Jaimes, and C. A. Coello Coello, "A survey on multiobjective evolutionary algorithms for the solution of the portfolio optimization problem and other finance and economics applications," *IEEE transactions on evolutionary computation*, vol. 17, no. 3, pp. 321–344, 2013.
- [6] Z.-H. Zhan, J.-Y. Li, and J. Zhang, "Evolutionary deep learning: A survey," *Neurocomputing*, vol. 483, pp. 42–58, 2022.
- [7] Y. Jin, *Multi-objective machine learning*. Springer Science & Business Media, 2006, vol. 16.
- [8] A. Telikani, A. Tahmassebi, W. Banzhaf, and A. H. Gandomi, "Evolutionary machine learning: A survey," *ACM Computing Surveys*, vol. 54, no. 8, 2022.
- [9] A. Mukhopadhyay, U. Maulik, S. Bandyopadhyay, and C. A. C. Coello, "A survey of multiobjective evolutionary algorithms for data mining: Part i," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 18, pp. 4–19, 2014.
- [10] X. Zhou, A. K. Qin, M. Gong, and K. C. Tan, "A survey on evolutionary construction of deep neural networks," *IEEE Transactions on Evolutionary Computation*, vol. 25, no. 5, pp. 894–912, 2021.
- [11] B. H. Nguyen, B. Xue, P. Andreae, and M. Zhang, "A hybrid evolutionary computation approach to inducing transfer classifiers for domain adaptation," *IEEE transactions on cybernetics*, vol. 51, no. 12, pp. 6319–6332, 2021.
- [12] P. P. Bonissone, R. Subbu, N. Eklund, and T. R. Kiehl, "Evolutionary algorithms+ domain knowledge= real-world evolutionary computation," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 3, pp. 256–280, 2006.
- [13] R. Salomon, "Evolutionary algorithms and gradient search: similarities and differences," *IEEE Transactions on Evolutionary Computation*, vol. 2, no. 2, pp. 45–55, 1998.
- [14] M. M. Drugan, "Reinforcement learning versus evolutionary computation: A survey on hybrid algorithms," *Swarm and evolutionary computation*, vol. 44, pp. 228–246, 2019.
- [15] R. Miikkulainen and S. Forrest, "A biological perspective on evolutionary computation," *Nature Machine Intelligence*, vol. 3, no. 1, pp. 9–15, 2021.
- [16] L. V. Santana-Quintero, A. A. Montano, and C. A. C. Coello, "A review of techniques for handling expensive functions in evolutionary multi-objective optimization," *Computational intelligence in expensive optimization problems*, pp. 29–59, 2010.
- [17] B. Li, J. Li, K. Tang, and X. Yao, "Many-objective evolutionary algorithms: A survey," *ACM Computing Surveys*, vol. 48, no. 1, pp. 1–35, 2015.
- [18] M. N. Omidvar, X. Li, and X. Yao, "A review of population-based metaheuristics for large-scale black-box global optimization-part i," *IEEE Transactions on Evolutionary Computation*, vol. 26, no. 5, pp. 802–822, 2021.
- [19] C. K. Goh and K. C. Tan, "An investigation on noisy environments in evolutionary multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 3, pp. 354–381, 2007.
- [20] T. Wei, S. Wang, J. Zhong, D. Liu, and J. Zhang, "A review on evolutionary multitask optimization: Trends and challenges," *IEEE Transactions on Evolutionary Computation*, vol. 26, no. 5, pp. 941–960, 2021.
- [21] Y. Zhai, Y.-S. Ong, and I. W. Tsang, "The emerging" big dimensionality," *IEEE Computational Intelligence Magazine*, vol. 9, no. 3, pp. 14–26, 2014.
- [22] Z.-H. Zhou, N. V. Chawla, Y. Jin, and G. J. Williams, "Big data opportunities and challenges: Discussions from data analytics perspectives [discussion forum]," *IEEE Computational intelligence magazine*, vol. 9, no. 4, pp. 62–74, 2014.
- [23] J. Zhang, Z.-h. Zhan, Y. Lin, N. Chen, Y.-j. Gong, J.-h. Zhong, H. S. Chung, Y. Li, and Y.-h. Shi, "Evolutionary computation meets machine learning: A survey," *IEEE Computational Intelligence Magazine*, vol. 6, no. 4, pp. 68–75, 2011.
- [24] D. E. Goldberg and J. H. Holland, "Genetic algorithms and machine learning," *Machine Learning*, vol. 3, no. 2-3, pp. 95–99, 1988.
- [25] L. Lin and M. Gen, "Hybrid evolutionary optimisation with learning for production scheduling: state-of-the-art survey on algorithms and applications," *International Journal of Production Research*, vol. 56, no. 1-2, pp. 193–223, 2018.
- [26] A. Liefvooghe, F. Daolio, S. Verel, B. Derbel, H. Aguirre, and K. Tanaka, "Landscape-aware performance prediction for evolutionary multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 6, pp. 1063–1077, 2020.
- [27] G. Li, Z. Wang, Q. Zhang, and J. Sun, "Offline and online objective reduction via gaussian mixture model clustering," *IEEE Transactions on Evolutionary Computation*, 2022.
- [28] K. Jiao, J. Chen, B. Xin, and L. Li, "A reference vector based multiobjective evolutionary algorithm with q-learning for operator adaptation," *Swarm and Evolutionary Computation*, vol. 76, p. 101225, 2023.
- [29] K. C. Tan, L. Feng, and M. Jiang, "Evolutionary transfer optimization—a new frontier in evolutionary computation research," *IEEE Computational Intelligence Magazine*, vol. 16, no. 1, pp. 22–33, 2021.
- [30] Z.-H. Zhan, L. Shi, K. C. Tan, and J. Zhang, "A survey on evolutionary computation for complex continuous optimization," *Artificial Intelligence Review*, vol. 55, no. 1, pp. 59–110, 2022.
- [31] J. A. Duro, D. K. Saxena, K. Deb, and Q. Zhang, "Machine learning based decision support for many-objective optimization problems," *Neurocomputing*, vol. 146, pp. 30–47, 2014.
- [32] J. Liang, X. Ban, K. Yu, B. Qu, K. Qiao, C. Yue, K. Chen, and K. C. Tan, "A survey on evolutionary constrained multi-objective optimization,"

- IEEE Transactions on Evolutionary Computation*, 2022.
- [33] Y. Tian, L. C. Si, X. Zhang, R. Cheng, C. He, K. C. Tan, and Y. Jin, "Evolutionary large-scale multi-objective optimization: A survey," *ACM Computing Surveys*, vol. 54, no. 8, pp. 1–34, 2021.
  - [34] R. Tanabe and H. Ishibuchi, "A review of evolutionary multimodal multi-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 1, pp. 193–200, 2020.
  - [35] A. Sinha, P. Malo, and K. Deb, "A review on bilevel optimization: From classical to evolutionary approaches and applications," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 2, pp. 276–295, 2017.
  - [36] Y. Jin, H. Wang, T. Chugh, D. Guo, and K. Miettinen, "Data-driven evolutionary optimization: An overview and case studies," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 3, 2019.
  - [37] Y.-S. Ong and A. Gupta, "Evolutionary multitasking: a computer science view of cognitive multitasking," *Cognitive Computation*, vol. 8, pp. 125–142, 2016.
  - [38] D. Yazdani, R. Cheng, D. Yazdani, J. Branke, Y. Jin, and X. Yao, "A survey of evolutionary continuous dynamic optimization over two decades-part a," *IEEE Transactions on Evolutionary Computation*, vol. 25, no. 4, pp. 609–629, 2021.
  - [39] M. Shakeri, E. Miah, A. Gupta, and Y.-S. Ong, "Scalable transfer evolutionary optimization: Coping with big task instances," *IEEE Transactions on Cybernetics*, 2022.
  - [40] S. Liu, Q. Lin, L. Feng, K.-C. Wong, and K. C. Tan, "Evolutionary multitasking for large-scale multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, 2022.
  - [41] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE transactions on evolutionary computation*, vol. 6, no. 2, pp. 182–197, 2002.
  - [42] K. Shang, H. Ishibuchi, L. He, and L. M. Pang, "A survey on the hypervolume indicator in evolutionary multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 25, no. 1, pp. 1–20, 2020.
  - [43] K. Li, "Decomposition multi-objective evolutionary optimization: From state-of-the-art to future opportunities," *arXiv preprint arXiv:2108.09588*, 2021.
  - [44] K. Sindhya, K. Miettinen, and K. Deb, "A hybrid framework for evolutionary multi-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 4, pp. 495–511, 2012.
  - [45] 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 transactions on evolutionary computation*, vol. 18, no. 4, pp. 577–601, 2013.
  - [46] H. Ishibuchi, Y. Setoguchi, H. Masuda, and Y. Nojima, "Performance of decomposition-based many-objective algorithms strongly depends on pareto front shapes," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 2, pp. 169–190, 2016.
  - [47] J. Deng and Q. Zhang, "Approximating hypervolume and hypervolume contributions using polar coordinate," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 5, pp. 913–918, 2019.
  - [48] K. Zhang, Z. Xu, S. Xie, and G. G. Yen, "Evolution strategy-based many-objective evolutionary algorithm through vector equilibrium," *IEEE transactions on cybernetics*, vol. 51, no. 11, pp. 5455–5467, 2020.
  - [49] S. Liu, Q. Lin, K.-C. Wong, Q. Li, and K. C. Tan, "Evolutionary large-scale multiobjective optimization: Benchmarks and algorithms," *IEEE Transactions on Evolutionary Computation*, 2021.
  - [50] S. Zapotecas-Martínez, C. A. C. Coello, H. E. Aguirre, and K. Tanaka, "A review of features and limitations of existing scalable multiobjective test suites," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 1, pp. 130–142, 2018.
  - [51] H. Wang, Y. Jin, and J. Doherty, "A generic test suite for evolutionary multifidelity optimization," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 6, pp. 836–850, 2017.
  - [52] J. J. Durillo, A. J. Nebro, C. A. C. Coello, J. García-Nieto, F. Luna, and E. Alba, "A study of multiobjective metaheuristics when solving parameter scalable problems," *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 4, pp. 618–635, 2010.
  - [53] Y. Jin and B. Sendhoff, "Pareto-based multiobjective machine learning: An overview and case studies," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 38, no. 3, pp. 397–415, 2008.
  - [54] A. Gupta, Y.-S. Ong, L. Feng, and K. C. Tan, "Multiobjective multifactorial optimization in evolutionary multitasking," *IEEE transactions on cybernetics*, vol. 47, no. 7, pp. 1652–1665, 2016.
  - [55] K. K. Bali, A. Gupta, Y.-S. Ong, and P. S. Tan, "Cognizant multitasking in multiobjective multifactorial evolution: Mo-mfea-ii," *IEEE transactions on cybernetics*, vol. 51, no. 4, pp. 1784–1796, 2020.
  - [56] L. Zhou, L. Feng, K. C. Tan, J. Zhong, Z. Zhu, K. Liu, and C. Chen, "Toward adaptive knowledge transfer in multifactorial evolutionary computation," *IEEE transactions on cybernetics*, vol. 51, no. 5, pp. 2563–2576, 2020.
  - [57] W. Li, G.-G. Wang, and A. H. Gandomi, "A survey of learning-based intelligent optimization algorithms," *Archives of Computational Methods in Engineering*, vol. 28, pp. 3781–3799, 2021.
  - [58] S. Bandaru, A. H. Ng, and K. Deb, "Data mining methods for knowledge discovery in multi-objective optimization: Part a-survey," *Expert Systems with Applications*, vol. 70, pp. 139–159, 2017.
  - [59] R. S. Michalski, "Learnable evolution model: Evolutionary processes guided by machine learning," *Machine learning*, vol. 38, pp. 9–40, 2000.
  - [60] R. Cheng, C. He, Y. Jin, and X. Yao, "Model-based evolutionary algorithms: a short survey," *Complex & Intelligent Systems*, vol. 4, no. 4, pp. 283–292, 2018.
  - [61] S. Chang, K. Yoo, J. Jang, and N. Kwak, "Self-evolutionary optimization for pareto front learning," *arXiv preprint arXiv:2110.03461*, 2021.
  - [62] T. M. Deist, M. Grewal, F. J. Dankers, T. Alderliesten, and P. A. Bosman, "Multi-objective learning to predict pareto fronts using hypervolume maximization," *arXiv preprint arXiv:2102.04523*, 2021.
  - [63] X. Chen, C. Shi, A. Zhou, B. Wu, and Z. Cai, "A decomposition based multiobjective evolutionary algorithm with semi-supervised classification," in *2017 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2017, pp. 797–804.
  - [64] J. Zhang, H. Ishibuchi, and L. He, "A classification-assisted environmental selection strategy for multiobjective optimization," *Swarm and Evolutionary Computation*, vol. 71, p. 101074, 2022.
  - [65] H.-L. Liu, F.-q. Gu, and Y.-m. Cheung, "T-moea/d: Moea/d with objective transform in multi-objective problems," in *2010 international conference of information science and management engineering*, vol. 2. IEEE, 2010, pp. 282–285.
  - [66] S. Zapotecas Martínez, V. A. Sosa Hernández, H. Aguirre, K. Tanaka, and C. A. Coello Coello, "Using a family of curves to approximate the pareto front of a multi-objective optimization problem," in *Parallel Problem Solving from Nature-PPSN XIII: 13th International Conference, Ljubljana, Slovenia, September 13-17, 2014. Proceedings 13*. Springer, 2014, pp. 682–691.
  - [67] S. Liu, Q. Yu, Q. Lin, and K. C. Tan, "An adaptive clustering-based evolutionary algorithm for many-objective optimization problems," *Information Sciences*, vol. 537, pp. 261–283, 2020.
  - [68] J. Luo, X. Sun, Y. Qi, and J. Xie, "Approximating the irregularly shaped pareto front of multi-objective reservoir flood control operation problem," *Applied Mathematical Modelling*, vol. 54, pp. 502–516, 2018.
  - [69] A. Zhou, Q. Zhang, and Y. Jin, "Approximating the set of pareto-optimal solutions in both the decision and objective spaces by an estimation of distribution algorithm," *IEEE transactions on evolutionary computation*, vol. 13, no. 5, pp. 1167–1189, 2009.
  - [70] M. Wu, K. Li, S. Kwong, Q. Zhang, and J. Zhang, "Learning to decompose: A paradigm for decomposition-based multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 3, pp. 376–390, 2018.
  - [71] Y. Tian, X. Zhang, R. Cheng, C. He, and Y. Jin, "Guiding evolutionary multiobjective optimization with generic front modeling," *IEEE transactions on cybernetics*, vol. 50, no. 3, pp. 1106–1119, 2018.
  - [72] J. Siwei, C. Zhihua, Z. Jie, and O. Yew-Soon, "Multiobjective optimization by decomposition with pareto-adaptive weight vectors," in *2011 Seventh international conference on natural computation*, vol. 3. IEEE, 2011, pp. 1260–1264.
  - [73] R. Wang, Q. Zhang, and T. Zhang, "Decomposition-based algorithms using pareto adaptive scalarizing methods," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 6, pp. 821–837, 2016.
  - [74] S. Liu, Q. Lin, K. C. Tan, M. Gong, and C. A. C. Coello, "A fuzzy decomposition-based multi/many-objective evolutionary algorithm," *IEEE transactions on cybernetics*, vol. 52, no. 5, pp. 3495–3509, 2020.

- [75] H. Xu, W. Zeng, X. Zeng, and G. G. Yen, "An evolutionary algorithm based on minkowski distance for many-objective optimization," *IEEE transactions on cybernetics*, vol. 49, no. 11, pp. 3968–3979, 2018.
- [76] A. G. Hernández-Díaz, L. V. Santana-Quintero, C. A. C. Coello, and J. Molina, "Pareto-adaptive  $\gamma$ -dominance," *Evolutionary computation*, vol. 15, no. 4, pp. 493–517, 2007.
- [77] C. M. Rebello, M. A. Martins, D. D. Santana, A. E. Rodrigues, J. M. Loureiro, A. M. Ribeiro, and I. B. Nogueira, "From a pareto front to pareto regions: A novel standpoint for multiobjective optimization," *Mathematics*, vol. 9, no. 24, p. 3152, 2021.
- [78] Y. Tian, L. Si, X. Zhang, K. C. Tan, and Y. Jin, "Local model-based pareto front estimation for multiobjective optimization," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 53, no. 1, pp. 623–634, 2022.
- [79] Y. Hua, Q. Liu, K. Hao, and Y. Jin, "A survey of evolutionary algorithms for multi-objective optimization problems with irregular pareto fronts," *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 2, pp. 303–318, 2021.
- [80] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach," *IEEE transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257–271, 1999.
- [81] S. Liu, Q. Lin, K.-C. Wong, C. A. C. Coello, J. Li, Z. Ming, and J. Zhang, "A self-guided reference vector strategy for many-objective optimization," *IEEE Transactions on Cybernetics*, vol. 52, no. 2, pp. 1164–1178, 2020.
- [82] R. Wang, Z. Zhou, H. Ishibuchi, T. Liao, and T. Zhang, "Localized weighted sum method for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 1, pp. 3–18, 2016.
- [83] Y. Yuan, H. Xu, B. Wang, and X. Yao, "A new dominance relation-based evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 1, pp. 16–37, 2015.
- [84] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff, "A reference vector guided evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 5, pp. 773–791, 2016.
- [85] L. Cai, S. Qu, Y. Yuan, and X. Yao, "A clustering-ranking method for many-objective optimization," *Applied Soft Computing*, vol. 35, pp. 681–694, 2015.
- [86] Q. Lin, S. Liu, K.-C. Wong, M. Gong, C. A. C. Coello, J. Chen, and J. Zhang, "A clustering-based evolutionary algorithm for many-objective optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 3, pp. 391–405, 2018.
- [87] J. Sun, H. Zhang, A. Zhou, Q. Zhang, K. Zhang, Z. Tu, and K. Ye, "Learning from a stream of nonstationary and dependent data in multi-objective evolutionary optimization," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 4, pp. 541–555, 2018.
- [88] S. Liu, Q. Lin, K.-C. Wong, L. Ma, C. A. C. Coello, and D. Gong, "A novel multi-objective evolutionary algorithm with dynamic decomposition strategy," *Swarm and Evolutionary Computation*, vol. 48, pp. 182–200, 2019.
- [89] Y. Hua, Y. Jin, and K. Hao, "A clustering-based adaptive evolutionary algorithm for multiobjective optimization with irregular pareto fronts," *IEEE Transactions on Cybernetics*, vol. 49, no. 7, pp. 2758–2770, 2018.
- [90] R. Denysiuk, L. Costa, and I. Espírito Santo, "Clustering-based selection for evolutionary many-objective optimization," in *Parallel Problem Solving from Nature—PPSN XIII: 13th International Conference, Ljubljana, Slovenia, September 13–17, 2014. Proceedings 13*. Springer, 2014, pp. 538–547.
- [91] L. A. Bejarano, H. E. Espitia, and C. E. Montenegro, "Clustering analysis for the pareto optimal front in multi-objective optimization," *Computation*, vol. 10, no. 3, p. 37, 2022.
- [92] H. Li, J. Sun, D. Meng, and Q. Zhang, "A multiobjective approach based on gaussian mixture clustering for sparse reconstruction," *IEEE Access*, vol. 7, pp. 22 684–22 697, 2019.
- [93] O. Kramer and H. Danielsiek, "DbSCAN-based multi-objective niching to approximate equivalent pareto-subsets," in *Proceedings of the 12th annual conference on Genetic and evolutionary computation*, 2010, pp. 503–510.
- [94] H. Ge, M. Zhao, L. Sun, Z. Wang, G. Tan, Q. Zhang, and C. P. Chen, "A many-objective evolutionary algorithm with two interacting processes: Cascade clustering and reference point incremental learning," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 4, pp. 572–586, 2018.
- [95] S. Liu, J. Zheng, Q. Lin, and K. C. Tan, "Evolutionary multi and many-objective optimization via clustering for environmental selection," *Information Sciences*, vol. 578, pp. 930–949, 2021.
- [96] Y. Liu, D. Gong, J. Sun, and Y. Jin, "A many-objective evolutionary algorithm using a one-by-one selection strategy," *IEEE Transactions on Cybernetics*, vol. 47, no. 9, pp. 2689–2702, 2017.
- [97] M. Pal, S. Saha, and S. Bandyopadhyay, "Decor: differential evolution using clustering based objective reduction for many-objective optimization," *Information Sciences*, vol. 423, pp. 200–218, 2018.
- [98] D. Brockhoff and E. Zitzler, "Improving hypervolume-based multiobjective evolutionary algorithms by using objective reduction methods," in *2007 IEEE congress on evolutionary computation*. IEEE, 2007, pp. 2086–2093.
- [99] Y. Yuan, Y.-S. Ong, A. Gupta, and H. Xu, "Objective reduction in many-objective optimization: evolutionary multiobjective approaches and comprehensive analysis," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 2, pp. 189–210, 2017.
- [100] K. Deb, D. Saxena *et al.*, "Searching for pareto-optimal solutions through dimensionality reduction for certain large-dimensional multi-objective optimization problems," in *Proceedings of the world congress on computational intelligence (WCCI-2006)*, 2006, pp. 3352–3360.
- [101] D. K. Saxena, J. A. Duro, A. Tiwari, K. Deb, and Q. Zhang, "Objective reduction in many-objective optimization: Linear and nonlinear algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 1, pp. 77–99, 2012.
- [102] L. Ma, R. Wang, S. Chen, S. Cheng, X. Wang, Z. Lin, Y. Shi, and M. Huang, "A novel many-objective evolutionary algorithm based on transfer matrix with kriging model," *Information Sciences*, vol. 509, pp. 437–456, 2020.
- [103] D. Brockhoff and E. Zitzler, "Objective reduction in evolutionary multiobjective optimization: Theory and applications," *Evolutionary computation*, vol. 17, no. 2, pp. 135–166, 2009.
- [104] F. Gu and Y.-M. Cheung, "Self-organizing map-based weight design for decomposition-based many-objective evolutionary algorithm," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 2, pp. 211–225, 2017.
- [105] Q. Liu, Y. Jin, M. Heiderich, T. Rodemann, and G. Yu, "An adaptive reference vector-guided evolutionary algorithm using growing neural gas for many-objective optimization of irregular problems," *IEEE Transactions on Cybernetics*, vol. 52, no. 5, pp. 2698–2711, 2020.
- [106] Y. Liu, H. Ishibuchi, N. Masuyama, and Y. Nojima, "Adapting reference vectors and scalarizing functions by growing neural gas to handle irregular pareto fronts," *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 3, pp. 439–453, 2019.
- [107] K. Li, T. Zhang, and R. Wang, "Deep reinforcement learning for multiobjective optimization," *IEEE transactions on cybernetics*, vol. 51, no. 6, pp. 3103–3114, 2020.
- [108] L. Ma, N. Li, Y. Guo, X. Wang, S. Yang, M. Huang, and H. Zhang, "Learning to optimize: reference vector reinforcement learning adaption to constrained many-objective optimization of industrial copper burdening system," *IEEE Transactions on Cybernetics*, 2021.
- [109] L. Cheng, Q. Tang, L. Zhang, and Z. Zhang, "Multi-objective q-learning-based hyper-heuristic with bi-criteria selection for energy-aware mixed shop scheduling," *Swarm and Evolutionary Computation*, vol. 69, p. 100985, 2022.
- [110] Y. Zhang, J. Wang, Z. Zhang, and Y. Zhou, "Modrl/d-el: multiobjective deep reinforcement learning with evolutionary learning for multiobjective optimization," in *2021 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2021, pp. 1–8.
- [111] Z. Zhang, Z. Wu, H. Zhang, and J. Wang, "Meta-learning-based deep reinforcement learning for multiobjective optimization problems," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.
- [112] K. Deb and R. Datta, "Hybrid evolutionary multi-objective optimization and analysis of machining operations," *Engineering Optimization*, vol. 44, no. 6, pp. 685–706, 2012.
- [113] M. Hauschild and M. Pelikan, "An introduction and survey of estimation of distribution algorithms," *Swarm and evolutionary computation*,



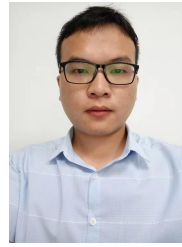
- vol. 1, no. 3, pp. 111–128, 2011.
- [114] W. Dong, T. Chen, P. Tiño, and X. Yao, “Scaling up estimation of distribution algorithms for continuous optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 6, pp. 797–822, 2013.
- [115] M. Laumanns and J. Ocenasek, “Bayesian optimization algorithms for multi-objective optimization,” *Lecture Notes in Computer Science*, vol. 2439, pp. 298–307, 2002.
- [116] X. Zhong and W. Li, “A decision-tree-based multi-objective estimation of distribution algorithm,” in *2007 International Conference on Computational Intelligence and Security (CIS 2007)*. IEEE, 2007, pp. 114–11.
- [117] H. Tang, V. A. Shim, K. C. Tan, and J. Y. Chia, “Restricted boltzmann machine based algorithm for multi-objective optimization,” in *IEEE congress on evolutionary computation*. IEEE, 2010, pp. 1–8.
- [118] L. Martí, J. García, A. Berlanga, C. A. C. Coello, and J. M. Molina, “Mb-gng: Addressing drawbacks in multi-objective optimization estimation of distribution algorithms,” *Operations Research Letters*, vol. 39, no. 2, pp. 150–154, 2011.
- [119] K. Li and S. Kwong, “A general framework for evolutionary multi-objective optimization via manifold learning,” *Neurocomputing*, vol. 146, pp. 65–74, 2014.
- [120] C. W. Ahn and R. S. Ramakrishna, “Multiobjective real-coded bayesian optimization algorithm revisited: diversity preservation,” in *Proceedings of the 9th annual conference on Genetic and evolutionary computation*, 2007, pp. 593–600.
- [121] M. Pelikan, K. Sastry, and D. E. Goldberg, “Multiobjective hboa, clustering, and scalability,” in *Proceedings of the 7th annual conference on Genetic and evolutionary computation*, 2005, pp. 663–670.
- [122] Q. Zhang, A. Zhou, and Y. Jin, “Rm-meda: A regularity model-based multiobjective estimation of distribution algorithm,” *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 1, pp. 41–63, 2008.
- [123] Y. Li, X. Xu, P. Li, and L. Jiao, “Improved rm-meda with local learning,” *Soft Computing*, vol. 18, pp. 1383–1397, 2014.
- [124] Y. Wang, J. Xiang, and Z. Cai, “A regularity model-based multiobjective estimation of distribution algorithm with reducing redundant cluster operator,” *Applied Soft Computing*, vol. 12, no. 11, pp. 3526–3538, 2012.
- [125] Y. Sun, G. G. Yen, and Z. Yi, “Improved regularity model-based eda for many-objective optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 5, pp. 662–678, 2018.
- [126] J. Sun, H. Zhang, A. Zhou, Q. Zhang, and K. Zhang, “A new learning-based adaptive multi-objective evolutionary algorithm,” *Swarm and evolutionary computation*, vol. 44, pp. 304–319, 2019.
- [127] P. A. Bosman and D. Thierens, “Multi-objective optimization with diversity preserving mixture-based iterated density estimation evolutionary algorithms,” *International Journal of Approximate Reasoning*, vol. 31, no. 3, pp. 259–289, 2002.
- [128] H. Karshenas, R. Santana, C. Bielza, and P. Larrañaga, “Multiobjective estimation of distribution algorithm based on joint modeling of objectives and variables,” *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 519–542, 2013.
- [129] H. Karshenas, R. Santana, C. Bielza, and P. Larrañaga, “Multi-objective optimization with joint probabilistic modeling of objectives and variables,” in *Proceedings of the 6th international conference on Evolutionary multi-criterion optimization*, 2011, pp. 298–312.
- [130] L. R. Farias and A. F. Araújo, “Im-moea/d: An inverse modeling multi-objective evolutionary algorithm based on decomposition,” in *2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2021, pp. 462–467.
- [131] R. Cheng, Y. Jin, K. Narukawa, and B. Sendhoff, “A multiobjective evolutionary algorithm using gaussian process-based inverse modeling,” *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 6, pp. 838–856, 2015.
- [132] E. Guerrero-Pena and A. F. R. Araújo, “Multi-objective evolutionary algorithm with prediction in the objective space,” *Information Sciences*, vol. 501, pp. 293–316, 2019.
- [133] K. Sastry, D. E. Goldberg, and M. Pelikan, “Limits of scalability of multiobjective estimation of distribution algorithms,” in *2005 IEEE Congress on Evolutionary Computation*, vol. 3. IEEE, 2005, pp. 2217–2224.
- [134] H. Hong, K. Ye, M. Jiang, D. Cao, and K. C. Tan, “Solving large-scale multiobjective optimization via the probabilistic prediction model,” *Memetic Computing*, vol. 14, no. 2, pp. 165–177, 2022.
- [135] C. He, S. Huang, R. Cheng, K. C. Tan, and Y. Jin, “Evolutionary multiobjective optimization driven by generative adversarial networks (gans),” *IEEE transactions on cybernetics*, vol. 51, no. 6, pp. 3129–3142, 2020.
- [136] Z. Wang, H. Hong, K. Ye, G.-E. Zhang, M. Jiang, and K. C. Tan, “Manifold interpolation for large-scale multiobjective optimization via generative adversarial networks,” *IEEE Transactions on Neural Networks and Learning Systems*, 2021.
- [137] X. Ma, F. Liu, Y. Qi, M. Gong, M. Yin, L. Li, L. Jiao, and J. Wu, “Moea/d with opposition-based learning for multiobjective optimization problem,” *Neurocomputing*, vol. 146, pp. 48–64, 2014.
- [138] A. A. Ewees, M. Abd Elaziz, and D. Oliva, “A new multi-objective optimization algorithm combined with opposition-based learning,” *Expert Systems with Applications*, vol. 165, p. 113844, 2021.
- [139] Y. Tian, X. Zheng, X. Zhang, and Y. Jin, “Efficient large-scale multiobjective optimization based on a competitive swarm optimizer,” *IEEE Transactions on Cybernetics*, vol. 50, no. 8, pp. 3696–3708, 2019.
- [140] S. Liu, Q. Lin, Q. Li, and K. C. Tan, “A comprehensive competitive swarm optimizer for large-scale multiobjective optimization,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 52, no. 9, pp. 5829–5842, 2021.
- [141] K. Deb and A. Srinivasan, “Innovation: Innovating design principles through optimization,” in *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, 2006, pp. 1629–1636.
- [142] S. Mittal, D. K. Saxena, K. Deb, and E. D. Goodman, “A learning-based innovized progress operator for faster convergence in evolutionary multi-objective optimization,” *ACM Transactions on Evolutionary Learning and Optimization (TELO)*, vol. 2, no. 1, pp. 1–29, 2021.
- [143] S. Bandaru and K. Deb, “Automated discovery of vital knowledge from pareto-optimal solutions: First results from engineering design,” in *IEEE congress on evolutionary computation*. IEEE, 2010, pp. 1–8.
- [144] N. Hittomi, H. Bang, and D. Selva, “Extracting and applying knowledge with adaptive knowledge-driven optimization to architect an earth observing satellite system,” in *AIAA Information Systems-AIAA Infotech@Aerospace*, 2017, p. 0794.
- [145] A. Gaur and K. Deb, “Effect of size and order of variables in rules for multi-objective repair-based innovization procedure,” in *2017 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2017, pp. 2177–2184.
- [146] S. Liu, J. Li, Q. Lin, Y. Tian, and K. C. Tan, “Learning to accelerate evolutionary search for large-scale multiobjective optimization,” *IEEE Transactions on Evolutionary Computation*, 2022.
- [147] S. Mittal, D. K. Saxena, K. Deb, and E. D. Goodman, “Enhanced innovized progress operator for evolutionary multi-and many-objective optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 26, no. 5, pp. 961–975, 2021.
- [148] S. Mittal, D. K. Saxena, and K. Deb, “Learning-based multi-objective optimization through ann-assisted online innovization,” in *Proceedings of the 2020 genetic and evolutionary computation conference companion*, 2020, pp. 171–172.
- [149] S. Mittal, D. Saxena, K. Deb, and E. Goodman, “An ann-assisted repair operator for evolutionary multiobjective optimization,” *COIN Report No. 2020005*, 2020.
- [150] H. Zhang, S. Song, A. Zhou, and X.-Z. Gao, “A clustering based multi-objective evolutionary algorithm,” in *2014 IEEE congress on evolutionary computation (CEC)*. IEEE, 2014, pp. 723–730.
- [151] C. von Lüken, C. A. Brizuela, and B. Barán, “Clustering-based multipopulation approaches in moea/d for many-objective problems,” *Computational Optimization and Applications*, vol. 81, no. 3, pp. 789–828, 2022.
- [152] L. Pan, L. Li, R. Cheng, C. He, and K. C. Tan, “Manifold learning-inspired mating restriction for evolutionary multiobjective optimization with complicated pareto sets,” *IEEE Transactions on Cybernetics*, vol. 51, no. 6, pp. 3325–3337, 2019.
- [153] S. Wang, H. Zhang, Y. Zhang, and A. Zhou, “Adaptive population structure learning in evolutionary multi-objective optimization,” *Soft Computing*, vol. 24, no. 13, pp. 10025–10042, 2020.
- [154] X. Zhang, Y. Tian, R. Cheng, and Y. Jin, “A decision variable clustering-based evolutionary algorithm for large-scale many-objective

- optimization," *IEEE Transactions on evolutionary Computation*, vol. 22, no. 1, pp. 97–112, 2016.
- [155] L. Ma, M. Huang, S. Yang, R. Wang, and X. Wang, "An adaptive localized decision variable analysis approach to large-scale multiobjective and many-objective optimization," *IEEE Transactions on Cybernetics*, vol. 52, no. 7, pp. 6684–6696, 2021.
- [156] W. Du, W. Zhong, Y. Tang, W. Du, and Y. Jin, "High-dimensional robust multi-objective optimization for order scheduling: A decision variable classification approach," *IEEE transactions on industrial informatics*, vol. 15, no. 1, pp. 293–304, 2018.
- [157] R. Liu, R. Ren, J. Liu, and J. Liu, "A clustering and dimensionality reduction based evolutionary algorithm for large-scale multi-objective problems," *Applied Soft Computing*, vol. 89, p. 106120, 2020.
- [158] Y. Tian, X. Zhang, C. Wang, and Y. Jin, "An evolutionary algorithm for large-scale sparse multiobjective optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 2, pp. 380–393, 2019.
- [159] S. Liu, M. Jiang, Q. Lin, and K. C. Tan, "Evolutionary large-scale multiobjective optimization via self-guided problem transformation," in *2022 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2022, pp. 1–8.
- [160] Y. Tian, C. Lu, X. Zhang, K. C. Tan, and Y. Jin, "Solving large-scale multiobjective optimization problems with sparse optimal solutions via unsupervised neural networks," *IEEE transactions on cybernetics*, vol. 51, no. 6, pp. 3115–3128, 2020.
- [161] Y. Tian, C. Lu, X. Zhang, F. Cheng, and Y. Jin, "A pattern mining-based evolutionary algorithm for large-scale sparse multiobjective optimization problems," *IEEE Transactions on Cybernetics*, vol. 52, no. 7, pp. 6784–6797, 2020.
- [162] C. Ying, J. Liu, K. Wu, and C. Wang, "A multiobjective evolutionary approach for solving large-scale network reconstruction problems via logistic principal component analysis," *IEEE Transactions on Cybernetics*, 2021.
- [163] X. Liu, Y. Du, M. Jiang, and X. Zeng, "Multiobjective particle swarm optimization based on network embedding for complex network community detection," *IEEE Transactions on Computational Social Systems*, vol. 7, no. 2, pp. 437–449, 2020.
- [164] C. Wang, R. Xu, J. Qiu, and X. Zhang, "Adaboost-inspired multi-operator ensemble strategy for multi-objective evolutionary algorithms," *Neurocomputing*, vol. 384, pp. 243–255, 2020.
- [165] J. Zhang, H. S.-H. Chung, and W.-L. Lo, "Clustering-based adaptive crossover and mutation probabilities for genetic algorithms," *IEEE Transactions on evolutionary Computation*, vol. 11, no. 3, pp. 326–335, 2007.
- [166] N. Hitomi and D. Selva, "A classification and comparison of credit assignment strategies in multiobjective adaptive operator selection," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 2, pp. 294–314, 2016.
- [167] A. Eiben, M. Horvath, W. Kowalczyk, and M. Schut, "Reinforcement learning for online control of evolutionary algorithms," *Engineering Self-Organising Systems*, pp. 151–160, 2007.
- [168] Y. Tian, X. Li, H. Ma, X. Zhang, K. C. Tan, and Y. Jin, "Deep reinforcement learning based adaptive operator selection for evolutionary multi-objective optimization," *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2022.
- [169] J.-J. Wang and L. Wang, "A cooperative memetic algorithm with learning-based agent for energy-aware distributed hybrid flow-shop scheduling," *IEEE Transactions on Evolutionary Computation*, vol. 26, no. 3, pp. 461–475, 2021.
- [170] H. Zhao and C. Zhang, "An online-learning-based evolutionary many-objective algorithm," *Information Sciences*, vol. 509, pp. 1–21, 2020.
- [171] P. A. Consoli, Y. Mei, L. L. Minku, and X. Yao, "Dynamic selection of evolutionary operators based on online learning and fitness landscape analysis," *Soft Computing*, vol. 20, pp. 3889–3914, 2016.
- [172] H. Zhao and C. Zhang, "A decomposition-based many-objective artificial bee colony algorithm with reinforcement learning," *Applied Soft Computing*, vol. 86, p. 105879, 2020.
- [173] Z. Li, L. Shi, C. Yue, Z. Shang, and B. Qu, "Differential evolution based on reinforcement learning with fitness ranking for solving multimodal multiobjective problems," *Swarm and Evolutionary Computation*, vol. 49, pp. 234–244, 2019.
- [174] Y. Huang, W. Li, F. Tian, and X. Meng, "A fitness landscape ruggedness multiobjective differential evolution algorithm with a reinforcement learning strategy," *Applied Soft Computing*, vol. 96, p. 106693, 2020.
- [175] Y. Sakurai, K. Takada, T. Kawabe, and S. Tsuruta, "A method to control parameters of evolutionary algorithms by using reinforcement learning," in *2010 sixth international conference on signal-image technology and internet based systems*. IEEE, 2010, pp. 74–79.
- [176] J. Sun, X. Liu, T. Bäck, and Z. Xu, "Learning adaptive differential evolution algorithm from optimization experiences by policy gradient," *IEEE Transactions on Evolutionary Computation*, vol. 25, no. 4, pp. 666–680, 2021.
- [177] A. Kaur and K. Kumar, "A reinforcement learning based evolutionary multi-objective optimization algorithm for spectrum allocation in cognitive radio networks," *Physical Communication*, vol. 43, p. 101196, 2020.
- [178] T. C. Bora, V. C. Mariani, and L. dos Santos Coelho, "Multi-objective optimization of the environmental-economic dispatch with reinforcement learning based on non-dominated sorting genetic algorithm," *Applied Thermal Engineering*, vol. 146, pp. 688–700, 2019.
- [179] J.-Y. Li, Z.-H. Zhan, and J. Zhang, "Evolutionary computation for expensive optimization: A survey," *Machine Intelligence Research*, vol. 19, no. 1, pp. 3–23, 2022.
- [180] J. Knowles, "Parego: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 1, pp. 50–66, 2006.
- [181] L. Pan, C. He, Y. Tian, H. Wang, X. Zhang, and Y. Jin, "A classification-based surrogate-assisted evolutionary algorithm for expensive many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 1, pp. 74–88, 2018.
- [182] Q. Zhang, W. Liu, E. Tsang, and B. Virginas, "Expensive multiobjective optimization by moea/d with gaussian process model," *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 3, pp. 456–474, 2009.
- [183] T. Chugh, Y. Jin, K. Miettinen, J. Hakanen, and K. Sindhya, "A surrogate-assisted reference vector guided evolutionary algorithm for computationally expensive many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 1, pp. 129–142, 2016.
- [184] Z. Song, H. Wang, C. He, and Y. Jin, "A kriging-assisted two-archive evolutionary algorithm for expensive many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 25, no. 6, pp. 1013–1027, 2021.
- [185] D. Guo, X. Wang, K. Gao, Y. Jin, J. Ding, and T. Chai, "Evolutionary optimization of high-dimensional multiobjective and many-objective expensive problems assisted by a dropout neural network," *IEEE transactions on systems, man, and cybernetics: systems*, vol. 52, no. 4, pp. 2084–2097, 2021.
- [186] J. Lin, C. He, and R. Cheng, "Adaptive dropout for high-dimensional expensive multiobjective optimization," *Complex & Intelligent Systems*, vol. 8, no. 1, pp. 271–285, 2022.
- [187] D. Guo, Y. Jin, J. Ding, and T. Chai, "Heterogeneous ensemble-based infill criterion for evolutionary multiobjective optimization of expensive problems," *IEEE transactions on cybernetics*, vol. 49, no. 3, pp. 1012–1025, 2018.
- [188] A. Habib, H. K. Singh, T. Chugh, T. Ray, and K. Miettinen, "A multiple surrogate assisted decomposition-based evolutionary algorithm for expensive multi/many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 6, pp. 1000–1014, 2019.
- [189] Q. Lin, X. Wu, L. Ma, J. Li, M. Gong, and C. A. C. Coello, "An ensemble surrogate-based framework for expensive multiobjective evolutionary optimization," *IEEE Transactions on Evolutionary Computation*, vol. 26, no. 4, pp. 631–645, 2021.
- [190] D. Zhan, Y. Cheng, and J. Liu, "Expected improvement matrix-based infill criteria for expensive multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 6, pp. 956–975, 2017.
- [191] J. Tian, Y. Tan, J. Zeng, C. Sun, and Y. Jin, "Multiobjective infill criterion driven gaussian process-assisted particle swarm optimization of high-dimensional expensive problems," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 3, pp. 459–472, 2018.
- [192] Y. Jin, "A comprehensive survey of fitness approximation in evolutionary computation," *Soft computing*, vol. 9, no. 1, pp. 3–12, 2005.
- [193] Y. S. Liao, K. C. Tan, J. Hu, X. Qiu, and S. B. Gee, "Machine learning

- enhanced multi-objective evolutionary algorithm based on decomposition,” in *14th International Conference on Intelligent Data Engineering and Automated Learning, IDEAL 2013*. Springer Berlin Heidelberg, 2013, pp. 553–560.
- [194] A. A. Rahat, R. M. Everson, and J. E. Fieldsend, “Alternative infill strategies for expensive multi-objective optimisation,” in *Proceedings of the genetic and evolutionary computation conference*, 2017, pp. 873–880.
- [195] W. Ponweiser, T. Wagner, D. Biermann, and M. Vincze, “Multiobjective optimization on a limited budget of evaluations using model-assisted  $\{S\}$ -metric selection,” *Parallel Problem Solving from Nature-PPSN X*, pp. 784–794, 2008.
- [196] V. Picheny, “Multiobjective optimization using gaussian process emulators via stepwise uncertainty reduction,” *Statistics and Computing*, vol. 25, no. 6, pp. 1265–1280, 2015.
- [197] C.-W. Seah, Y.-S. Ong, I. W. Tsang, and S. Jiang, “Pareto rank learning in multi-objective evolutionary algorithms,” in *2012 IEEE Congress on Evolutionary Computation*. IEEE, 2012, pp. 1–8.
- [198] H. Hao, J. Zhang, X. Lu, and A. Zhou, “Binary relation learning and classifying for preselection in evolutionary algorithms,” *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 6, pp. 1125–1139, 2020.
- [199] T. Chugh, K. Sindhya, J. Hakanen, and K. Miettinen, “A survey on handling computationally expensive multiobjective optimization problems with evolutionary algorithms,” *Soft Computing*, vol. 23, pp. 3137–3166, 2019.
- [200] S. Bandaru, A. H. Ng, and K. Deb, “On the performance of classification algorithms for learning pareto-dominance relations,” in *2014 IEEE Congress on evolutionary computation (CEC)*. IEEE, 2014, pp. 1139–1146.
- [201] Y. Yuan and W. Banzhaf, “Expensive multiobjective evolutionary optimization assisted by dominance prediction,” *IEEE Transactions on Evolutionary Computation*, vol. 26, no. 1, pp. 159–173, 2021.
- [202] H. Hao, A. Zhou, H. Qian, and H. Zhang, “Expensive multiobjective optimization by relation learning and prediction,” *IEEE Transactions on Evolutionary Computation*, vol. 26, no. 5, pp. 1157–1170, 2022.
- [203] J. Zhang, A. Zhou, K. Tang, and G. Zhang, “Preselection via classification: A case study on evolutionary multiobjective optimization,” *Information Sciences*, vol. 465, pp. 388–403, 2018.
- [204] X. Lin, Q. Zhang, and S. Kwong, “A decomposition based multiobjective evolutionary algorithm with classification,” in *2016 IEEE congress on evolutionary computation (CEC)*. IEEE, 2016, pp. 3292–3299.
- [205] T. Sonoda and M. Nakata, “Multiple classifiers-assisted evolutionary algorithm based on decomposition for high-dimensional multiobjective problems,” *IEEE Transactions on Evolutionary Computation*, vol. 26, no. 6, pp. 1581–1595, 2022.
- [206] A. Gupta, Y.-S. Ong, and L. Feng, “Insights on transfer optimization: Because experience is the best teacher,” *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 1, pp. 51–64, 2017.
- [207] D. Xu, M. Jiang, W. Hu, S. Li, R. Pan, and G. G. Yen, “An online prediction approach based on incremental support vector machine for dynamic multiobjective optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 26, no. 4, pp. 690–703, 2021.
- [208] C. Raquel and X. Yao, “Dynamic multi-objective optimization: a survey of the state-of-the-art,” in *Evolutionary computation for dynamic optimization problems*. Springer, 2013, pp. 85–106.
- [209] Q. Chen, J. Ding, S. Yang, and T. Chai, “A novel evolutionary algorithm for dynamic constrained multiobjective optimization problems,” *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 4, pp. 792–806, 2019.
- [210] S. Jiang, J. Zou, S. Yang, and X. Yao, “Evolutionary dynamic multi-objective optimisation: A survey,” *ACM Computing Surveys*, vol. 55, no. 4, pp. 1–47, 2022.
- [211] X. Fu and J. Sun, “A new learning based dynamic multi-objective optimisation evolutionary algorithm,” in *2017 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2017, pp. 341–348.
- [212] J. Zhang, W. Zhou, X. Chen, W. Yao, and L. Cao, “Multisource selective transfer framework in multiobjective optimization problems,” *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 3, pp. 424–438, 2019.
- [213] B. Da, A. Gupta, and Y.-S. Ong, “Curbing negative influences online for seamless transfer evolutionary optimization,” *IEEE transactions on cybernetics*, vol. 49, no. 12, pp. 4365–4378, 2018.
- [214] M. Jiang, Z. Huang, L. Qiu, W. Huang, and G. G. Yen, “Transfer learning-based dynamic multiobjective optimization algorithms,” *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 4, pp. 501–514, 2017.
- [215] Y. Hu, J. Zheng, S. Jiang, S. Yang, and J. Zou, “Handling dynamic multiobjective optimization environments via layered prediction and subspace-based diversity maintenance,” *IEEE Transactions on Cybernetics*, 2021.
- [216] X. Ma, J. Yang, H. Sun, Z. Hu, and L. Wei, “Feature information prediction algorithm for dynamic multi-objective optimization problems,” *European Journal of Operational Research*, vol. 295, no. 3, pp. 965–981, 2021.
- [217] X. Ma, Y. Zheng, Z. Zhu, X. Li, L. Wang, Y. Qi, and J. Yang, “Improving evolutionary multitasking optimization by leveraging inter-task gene similarity and mirror transformation,” *IEEE Computational Intelligence Magazine*, vol. 16, no. 4, pp. 38–53, 2021.
- [218] G. Ruan, L. L. Minku, S. Menzel, B. Sendhoff, and X. Yao, “When and how to transfer knowledge in dynamic multi-objective optimization,” in *2019 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2019, pp. 2034–2041.
- [219] S. Jiang and S. Yang, “A steady-state and generational evolutionary algorithm for dynamic multiobjective optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 1, pp. 65–82, 2016.
- [220] D. Yazdani, M. N. Omidvar, J. Branke, T. T. Nguyen, and X. Yao, “Scaling up dynamic optimization problems: A divide-and-conquer approach,” *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 1, pp. 1–15, 2019.
- [221] X. Xue, C. Yang, Y. Hu, K. Zhang, Y.-M. Cheung, L. Song, and K. C. Tan, “Evolutionary sequential transfer optimization for objective-heterogeneous problems,” *IEEE Transactions on Evolutionary Computation*, vol. 26, no. 6, pp. 1424–1438, 2021.
- [222] Z. Liang, X. Xu, L. Liu, Y. Tu, and Z. Zhu, “Evolutionary many-task optimization based on multisource knowledge transfer,” *IEEE Transactions on Evolutionary Computation*, vol. 26, no. 2, pp. 319–333, 2021.
- [223] J. Lin, H.-L. Liu, B. Xue, M. Zhang, and F. Gu, “Multiobjective multitasking optimization based on incremental learning,” *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 5, pp. 824–838, 2019.
- [224] J. Lin, H.-L. Liu, K. C. Tan, and F. Gu, “An effective knowledge transfer approach for multiobjective multitasking optimization,” *IEEE transactions on cybernetics*, vol. 51, no. 6, pp. 3238–3248, 2020.
- [225] H. Chen, H.-L. Liu, F. Gu, and K. C. Tan, “A multi-objective multi-task optimization algorithm using transfer rank,” *IEEE Transactions on Evolutionary Computation*, 2022.
- [226] H. Han, X. Bai, H. Han, Y. Hou, and J. Qiao, “Self-adjusting multitask particle swarm optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 26, no. 1, pp. 145–158, 2021.
- [227] X. Wang, Z. Dong, L. Tang, and Q. Zhang, “Multiobjective multitask optimization-neighborhood as a bridge for knowledge transfer,” *IEEE Transactions on Evolutionary Computation*, 2022.
- [228] L. Feng, Y.-S. Ong, S. Jiang, and A. Gupta, “Autoencoding evolutionary search with learning across heterogeneous problems,” *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 5, pp. 760–772, 2017.
- [229] L. Feng, Y.-S. Ong, A.-H. Tan, and I. W. Tsang, “Memes as building blocks: a case study on evolutionary optimization+ transfer learning for routing problems,” *Memetic Computing*, vol. 7, pp. 159–180, 2015.
- [230] L. Feng, W. Zhou, W. Liu, Y.-S. Ong, and K. C. Tan, “Solving dynamic multiobjective problem via autoencoding evolutionary search,” *IEEE Transactions on Cybernetics*, vol. 52, no. 5, pp. 2649–2662, 2020.
- [231] L. Zhou, L. Feng, A. Gupta, and Y.-S. Ong, “Learnable evolutionary search across heterogeneous problems via kernelized autoencoding,” *IEEE Transactions on Evolutionary Computation*, vol. 25, no. 3, pp. 567–581, 2021.
- [232] W. Zhou, L. Feng, K. C. Tan, M. Jiang, and Y. Liu, “Evolutionary search with multiview prediction for dynamic multiobjective optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 26, no. 5, pp. 911–925, 2021.
- [233] Z. Liang, H. Dong, C. Liu, W. Liang, and Z. Zhu, “Evolutionary multitasking for multiobjective optimization with subspace alignment and adaptive differential evolution,” *IEEE Transactions on Cybernetics*,

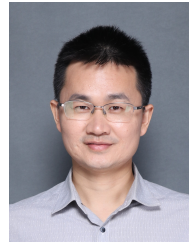
- vol. 52, no. 4, pp. 2096–2109, 2020.
- [234] Z. Tang, M. Gong, Y. Wu, W. Liu, and Y. Xie, “Regularized evolutionary multitask optimization: Learning to intertask transfer in aligned subspace,” *IEEE Transactions on Evolutionary Computation*, vol. 25, no. 2, pp. 262–276, 2020.
- [235] Z. Chen, Y. Zhou, X. He, and J. Zhang, “Learning task relationships in evolutionary multitasking for multiobjective continuous optimization,” *IEEE Transactions on Cybernetics*, vol. 52, no. 6, pp. 5278–5289, 2020.
- [236] M. Jiang, Z. Wang, L. Qiu, S. Guo, X. Gao, and K. C. Tan, “A fast dynamic evolutionary multiobjective algorithm via manifold transfer learning,” *IEEE Transactions on Cybernetics*, vol. 51, no. 7, pp. 3417–3428, 2020.
- [237] Z. Tang, M. Gong, Y. Wu, A. Qin, and K. C. Tan, “A multifactorial optimization framework based on adaptive intertask coordinate system,” *IEEE Transactions on Cybernetics*, vol. 52, no. 7, pp. 6745–6758, 2021.
- [238] R. Lim, L. Zhou, A. Gupta, Y.-S. Ong, and A. N. Zhang, “Solution representation learning in multi-objective transfer evolutionary optimization,” *IEEE Access*, vol. 9, pp. 41 844–41 860, 2021.
- [239] J. Zou, Q. Li, S. Yang, J. Zheng, Z. Peng, and T. Pei, “A dynamic multiobjective evolutionary algorithm based on a dynamic evolutionary environment model,” *Swarm and evolutionary computation*, vol. 44, pp. 247–259, 2019.
- [240] Z. Liang, W. Liang, Z. Wang, X. Ma, L. Liu, and Z. Zhu, “Multiobjective evolutionary multitasking with two-stage adaptive knowledge transfer based on population distribution,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 52, no. 7, pp. 4457–4469, 2021.
- [241] X.-F. Liu, Z.-H. Zhan, T.-L. Gu, S. Kwong, Z. Lu, H. B.-L. Duh, and J. Zhang, “Neural network-based information transfer for dynamic optimization,” *IEEE transactions on neural networks and learning systems*, vol. 31, no. 5, pp. 1557–1570, 2019.
- [242] L. Cao, L. Xu, E. D. Goodman, C. Bao, and S. Zhu, “Evolutionary dynamic multiobjective optimization assisted by a support vector regression predictor,” *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 2, pp. 305–319, 2019.
- [243] J.-Y. Li, Z.-H. Zhan, K. C. Tan, and J. Zhang, “A meta-knowledge transfer-based differential evolution for multitask optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 26, no. 4, pp. 719–734, 2021.
- [244] R. Lim, A. Gupta, Y.-S. Ong, L. Feng, and A. N. Zhang, “Non-linear domain adaptation in transfer evolutionary optimization,” *Cognitive Computation*, vol. 13, pp. 290–307, 2021.
- [245] P. Liao, C. Sun, G. Zhang, and Y. Jin, “Multi-surrogate multi-tasking optimization of expensive problems,” *Knowledge-Based Systems*, vol. 205, p. 106262, 2020.
- [246] B. Li, G. Wu, Y. He, M. Fan, and W. Pedrycz, “An overview and experimental study of learning-based optimization algorithms for the vehicle routing problem,” *IEEE/CAA Journal of Automatica Sinica*, vol. 9, no. 7, pp. 1115–1138, 2022.
- [247] X. Wang, Y. Jin, S. Schmitt, and M. Olhofer, “Transfer learning for gaussian process assisted evolutionary bi-objective optimization for objectives with different evaluation times,” in *Proceedings of the 2020 genetic and evolutionary computation conference*, 2020, pp. 587–594.
- [248] —, “Transfer learning based co-surrogate assisted evolutionary bi-objective optimization for objectives with non-uniform evaluation times,” *Evolutionary computation*, vol. 30, no. 2, pp. 221–251, 2022.
- [249] J. Ding, C. Yang, Y. Jin, and T. Chai, “Generalized multitasking for evolutionary optimization of expensive problems,” *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 1, pp. 44–58, 2017.
- [250] Z. Chen, A. Gupta, L. Zhou, and Y.-S. Ong, “Scaling multiobjective evolution to large data with minions: A bayes-informed multitask approach,” *IEEE Transactions on Cybernetics*, 2022.
- [251] K. Li, R. Chen, and X. Yao, “A data-driven evolutionary transfer optimization for expensive problems in dynamic environments,” *arXiv preprint arXiv:2211.02879*, 2022.
- [252] X. Fan, K. Li, and K. C. Tan, “Surrogate assisted evolutionary algorithm based on transfer learning for dynamic expensive multi-objective optimisation problems,” in *2020 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2020, pp. 1–8.
- [253] A. T. W. Min, Y.-S. Ong, A. Gupta, and C.-K. Goh, “Multiproblem surrogates: Transfer evolutionary multiobjective optimization of computationally expensive problems,” *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 1, pp. 15–28, 2017.
- [254] J. Luo, L. Chen, X. Li, and Q. Zhang, “Novel multitask conditional neural-network surrogate models for expensive optimization,” *IEEE Transactions on Cybernetics*, vol. 52, no. 5, pp. 3984–3997, 2020.
- [255] J. Li, T. Sun, Q. Lin, M. Jiang, and K. C. Tan, “Reducing negative transfer learning via clustering for dynamic multiobjective optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 26, no. 5, pp. 1102–1116, 2022.
- [256] Y. Guo, H. Yang, M. Chen, J. Cheng, and D. Gong, “Ensemble prediction-based dynamic robust multi-objective optimization methods,” *Swarm and evolutionary computation*, vol. 48, pp. 156–171, 2019.
- [257] F. Zou, G. G. Yen, L. Tang, and C. Wang, “A reinforcement learning approach for dynamic multi-objective optimization,” *Information Sciences*, vol. 546, pp. 815–834, 2021.
- [258] A. Zhou, Y. Jin, and Q. Zhang, “A population prediction strategy for evolutionary dynamic multiobjective optimization,” *IEEE transactions on cybernetics*, vol. 44, no. 1, pp. 40–53, 2013.
- [259] H. Zhang, J. Ding, M. Jiang, K. C. Tan, and T. Chai, “Inverse gaussian process modeling for evolutionary dynamic multiobjective optimization,” *IEEE Transactions on Cybernetics*, vol. 52, no. 10, pp. 11 240–11 253, 2021.
- [260] S. B. Gee, K. C. Tan, and C. Alippi, “Solving multiobjective optimization problems in unknown dynamic environments: An inverse modeling approach,” *IEEE transactions on cybernetics*, vol. 47, no. 12, pp. 4223–4234, 2016.
- [261] T. Wei and J. Zhong, “Towards generalized resource allocation on evolutionary multitasking for multi-objective optimization,” *IEEE Computational Intelligence Magazine*, vol. 16, no. 4, pp. 20–37, 2021.
- [262] Y. Fang, F. Liu, M. Li, and H. Cui, “Domain generalization-based dynamic multiobjective optimization: A case study on disassembly line balancing,” *IEEE Transactions on Evolutionary Computation*, 2023.
- [263] Y. Guo, G. Chen, M. Jiang, D. Gong, and J. Liang, “A knowledge guided transfer strategy for evolutionary dynamic multiobjective optimization,” *IEEE Transactions on Evolutionary Computation*, 2022.
- [264] E. Osaba, J. Del Ser, A. D. Martinez, and A. Hussain, “Evolutionary multitask optimization: a methodological overview, challenges, and future research directions,” *Cognitive Computation*, vol. 14, no. 3, pp. 927–954, 2022.
- [265] E. Osaba, J. Del Ser, and P. N. Suganthan, “Evolutionary multitask optimization: Fundamental research questions, practices, and directions for the future,” *Swarm and Evolutionary Computation*, vol. 75, p. 101203, 2022.
- [266] Z.-H. Zhan, J.-Y. Li, S. Kwong, and J. Zhang, “Learning-aided evolution for optimization,” *IEEE Transactions on Evolutionary Computation*, 2022.
- [267] T. Chai, Y. Jin, and S. Bernhard, “Evolutionary complex engineering optimization: Opportunities and challenges,” *IEEE Computational Intelligence Magazine*, vol. 8, no. 3, pp. 12–15, 2013.
- [268] H. Ishibuchi, L. M. Pang, and K. Shang, “Difficulties in fair performance comparison of multi-objective evolutionary algorithms [research frontier],” *IEEE Computational Intelligence Magazine*, vol. 17, no. 1, pp. 86–101, 2022.
- [269] E. Osaba, E. Villar-Rodriguez, J. Del Ser, A. J. Nebro, D. Molina, A. LaTorre, P. N. Suganthan, C. A. C. Coello, and F. Herrera, “A tutorial on the design, experimentation and application of metaheuristic algorithms to real-world optimization problems,” *Swarm and Evolutionary Computation*, vol. 64, p. 100888, 2021.
- [270] J. Zhang, L. He, and H. Ishibuchi, “Dual fuzzy classifier-based evolutionary algorithm for expensive multiobjective optimization,” *IEEE Transactions on Evolutionary Computation*, 2022.
- [271] F. Li, L. Gao, and W. Shen, “Surrogate-assisted multi-objective evolutionary optimization with pareto front model-based local search method,” *IEEE Transactions on Cybernetics*, 2022.
- [272] T. Zhang, H. Wang, B. Yuan, Y. Jin, and X. Yao, “Surrogate-assisted evolutionary q-learning for black-box dynamic time-linkage optimization problems,” *IEEE Transactions on Evolutionary Computation*, 2022.
- [273] Q. Liu, R. Cheng, Y. Jin, M. Heiderich, and T. Rodemann, “Reference vector-assisted adaptive model management for surrogate-assisted many-objective optimization,” *IEEE Transactions on Systems, Man, and*

- Cybernetics: Systems*, vol. 52, no. 12, pp. 7760–7773, 2022.
- [274] Y. Zhang, X.-f. Ji, X.-z. Gao, D.-w. Gong, and X.-y. Sun, “Objective-constraint mutual-guided surrogate-based particle swarm optimization for expensive constrained multimodal problems,” *IEEE Transactions on Evolutionary Computation*, 2022.
- [275] S. Jiang, G. G. Yen, and Z. He, “A multi-scenario optimization evolutionary algorithm based on transfer framework,” *IEEE Transactions on Evolutionary Computation*, 2022.
- [276] J. Yi, J. Bai, H. He, W. Zhou, and L. Yao, “A multifactorial evolutionary algorithm for multitasking under interval uncertainties,” *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 5, pp. 908–922, 2020.
- [277] K. Qiao, K. Yu, B. Qu, J. Liang, H. Song, and C. Yue, “An evolutionary multitasking optimization framework for constrained multiobjective optimization problems,” *IEEE Transactions on Evolutionary Computation*, vol. 26, no. 2, pp. 263–277, 2022.
- [278] F. Ming, W. Gong, L. Wang, and L. Gao, “Constrained multi-objective optimization via multitasking and knowledge transfer,” *IEEE Transactions on Evolutionary Computation*, 2022.
- [279] Q. Liu, Y. Yan, P. Ligeti, and Y. Jin, “A secure federated data-driven evolutionary multi-objective optimization algorithm,” *arXiv preprint arXiv:2210.08295*, 2022.
- [280] X. Hao, R. Qu, and J. Liu, “A unified framework of graph-based evolutionary multitasking hyper-heuristic,” *IEEE Transactions on Evolutionary Computation*, vol. 25, no. 1, pp. 35–47, 2020.
- [281] X. Yan, H. Huang, Z. Hao, and J. Wang, “A graph-based fuzzy evolutionary algorithm for solving two-echelon vehicle routing problems,” *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 1, pp. 129–141, 2019.
- [282] W. Liu, R. Wang, T. Zhang, K. Li, W. Li, and H. Ishibuchi, “Hybridization of evolutionary algorithm and deep reinforcement learning for multi-objective orienteering optimization,” *IEEE Transactions on Evolutionary Computation*, 2022.
- [283] T. Bartz-Beielstein, C. Doerr, D. v. d. Berg, J. Bossek, S. Chandrasekaran, T. Eftimov, A. Fischbach, P. Kerschke, W. La Cava, M. Lopez-Ibanez *et al.*, “Benchmarking in optimization: Best practice and open issues,” *arXiv preprint arXiv:2007.03488*, 2020.
- [284] Z. Liu, H. Wang, and Y. Jin, “Performance indicator-based adaptive model selection for offline data-driven multiobjective evolutionary optimization,” *IEEE Transactions on Cybernetics*, 2022.
- [285] Z. Lu, R. Cheng, Y. Jin, K. C. Tan, and K. Deb, “Neural architecture search as multiobjective optimization benchmarks: Problem formulation and performance assessment,” *IEEE Transactions on Evolutionary Computation*, 2023.
- [286] B. H. Nguyen, B. Xue, and M. Zhang, “A constrained competitive swarm optimiser with an svm-based surrogate model for feature selection,” *IEEE Transactions on Evolutionary Computation*, 2022.
- [287] Y. Tan and K. Ding, “A survey on gpu-based implementation of swarm intelligence algorithms,” *IEEE transactions on cybernetics*, vol. 46, no. 9, pp. 2028–2041, 2015.
- [288] J.-Y. Li, Z.-H. Zhan, R.-D. Liu, C. Wang, S. Kwong, and J. Zhang, “Generation-level parallelism for evolutionary computation: A pipeline-based parallel particle swarm optimization,” *IEEE Transactions on Cybernetics*, vol. 51, no. 10, pp. 4848–4859, 2020.
- [289] Y. Huang, L. Feng, A. K. Qin, M. Chen, and K. C. Tan, “Toward large-scale evolutionary multitasking: A gpu-based paradigm,” *IEEE Transactions on Evolutionary Computation*, vol. 26, no. 3, pp. 585–598, 2021.
- [290] K. Li and R. Chen, “Batched data-driven evolutionary multi-objective optimization based on manifold interpolation,” *IEEE Transactions on Evolutionary Computation*, 2022.
- [291] H. Wang and Y. Jin, “A random forest-assisted evolutionary algorithm for data-driven constrained multiobjective combinatorial optimization of trauma systems,” *IEEE transactions on cybernetics*, vol. 50, no. 2, pp. 536–549, 2018.
- [292] W. Yi, R. Qu, L. Jiao, and B. Niu, “Automated design of metaheuristics using reinforcement learning within a novel general search framework,” *IEEE Transactions on Evolutionary Computation*, 2022.



**Songbai Liu** (Member IEEE) received the B.S. degree from Changsha University and the M.S. degree from Shenzhen University, China, in 2012 and 2018, respectively. He received the Ph.D. degree from Department of Computer Sciences, City University of Hong Kong, in 2022.

He is currently an assistant professor in College of Computer Science and Software Engineering, Shenzhen University. His research interests include evolutionary algorithms + machine learning, evolutionary large-scale optimization, and their applications.



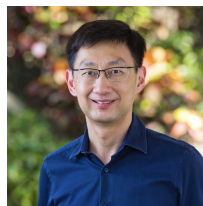
**Qiuzhen Lin** (Member IEEE) received the B.S. degree from Zhaoqing University and the M.S. degree from Shenzhen University, China, in 2007 and 2010, respectively. He received the Ph.D. degree from Department of Elec-tronic Engineering, City University of Hong Kong, Kowloon, Hong Kong, in 2014.

He is currently an associate professor in College of Computer Science and Software Engineering, Shenzhen University. He has published over sixty research papers since 2008. He is an Associate Editor of the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION and the IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTATIONAL INTELLIGENCE. His current research interests include artificial immune system, multi-objective optimization, and dynamic system.



**Jianqiang Li** (Member IEEE) received his B.S. and Ph.D. Degree in automation major from South China University of Technology, Guangzhou, China, in 2003 and 2008, respectively.

He is a professor at the College of Computer and Software Engineering of Shenzhen University. He led five projects of the National Natural Science Foundation and four projects of the Natural Science Foundation of Guangdong Province, China. His current research interests include robotics, embedded systems, and Internet of Things.



**Kay Chen Tan** (Fellow, IEEE) received the B.Eng. degree (First Class Hons.) and the Ph.D. degree from the University of Glasgow, U.K., in 1994 and 1997, respectively. He is currently a Chair Professor of the Department of Computing, The Hong Kong Polytechnic University. He was the Editor-in-Chief of IEEE Transactions on Evolutionary Computation, and currently serves on the Editorial Board member of 10+ journals. He is currently the Vice-President (Publications) of IEEE Computational Intelligence Society, an Honorary Professor at University of Nottingham in UK, and the Chief Co-Editor of Springer Book Series on Machine Learning: Foundations, Methodologies, and Applications. He is an IEEE Fellow.

Nottingham in UK, and the Chief Co-Editor of Springer Book Series on Machine Learning: Foundations, Methodologies, and Applications. He is an IEEE Fellow.