A Survey on Evolutionary Algorithms for Scalable Multiobjective Optimization
(Supplementary Document)

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

***Abstract*—This is the supplementary document of the paper titled "A Survey on Evolutionary Algorithms for Scalable Multiobjective Optimization", which is published on the journal of IEEE Transactions on Evolutionary Computation. In this supplementary document, an analysis of the challenges that scalable MOPs pose to traditional MOEAs is firstly proposed. Then, discusses on learnable discriminators via clustering are supplemented. Finally, the figures used to illustrate the ideal of evolutionary transfer optimization and domain adaptation are also proposed.**

1. 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** of candidate solutions to approximate the optima. As shown in Fig. 2 and Fig. 3, this process mainly includes three modules: 1) the evolutionary generator for reproduction that reproduces a new offspring population **Q** by exploring the search space; 2) the function evaluator to get a solution's objective vector by mapping it into the objective space; 3) the evolutionary discriminator for environmental selection that keeps the fittest solutions to the next generation population by differentiating elite and poor solutions from the combined population **P**+**Q**. The potential bottlenecks that limit a traditional MOEA's generalization ability in handling the MOPs with scaling-up complexity and dimensionality can be summarized as follows:

At first, the literature on EMO has overemphasized the role of environmental selection (i.e., the discriminator), neglecting a richer family of related components, e.g., genetic operators and search policies used in the generator. In this context, a plethora of environmental selection strategies has been developed, which can be roughly divided into four categories, i.e., Pareto-based [1], indicator-based [2], decomposition-based [3], and their hybridization [4] strategies. These selection strategies were designed to balance the convergence and diversity of the evolutionary population by customizing a discriminant criterion so that the solutions are comparable in the objective space. Concretely, Pareto-based MOEAs define a nondominated ranking scheme at first to sort all solutions into different fronts, in which the solutions arranged in the lower front have better convergence performance, followed by adopting a diversity maintenance method to enhance the diversity of solutions. Decomposition-based MOEAs divide an MOP into multiple scalar subproblems using a set of weight vectors and then define an aggregated fitness value for each solution. Indicator-based MOEAs measure the performance of solutions via a comprehensive performance indicator such as hypervolume that can reflect both the convergence and diversity contribution of this solution. Although these selection criteria have been demonstrated in effectively tackling MOPs with two or three objectives, their effectiveness is greatly discounted when used to solve MaOPs. To be specific, almost all solutions are ranked in the first front as they are mostly mutually non-dominated, resulting in insufficient selection pressure for Pareto-based strategies [5]. The effectiveness of decomposition-based strategies deteriorates severely because their pre-specified weight vectors cannot match well with the increasingly complex and irregular Pareto fronts (PFs) [6]. Similar issues could also exist in indicator-based strategies, e.g., solutions have biased distribution in the middle of a convex (or concave) PF due to their larger (or smaller) HV contributions than those on the borders. Besides, the cost of computing hypervolume (or others like it) grows exponentially as the number of objectives increases, which makes this criterion unsuitable for tackling MaOPs [7]. Thus, these traditional MOEAs have struggled to meet the challenges posed by scaling-up MOPs in the objective space.

Secondly, the reproduction of the offspring population via a generator is a search process for finding promising solutions in the variable space of an MOP. Proverbially, this search space expands exponentially with the number of variables increases, which requires the generator with growing search capabilities (e.g., accelerated convergence speed) to find the nearly optimal solution set within a limited computing budget. What gets trickier is that the structure and properties (e.g., multimodality, separability, and nonlinearity) of large-scale MOPs become incredibly complex. Concretely, with more variables in MOPs, the linkage and interaction between variables become more intricate, the contribution of different variables to multiple objectives becomes more imbalanced, and the overlap between shared variables that define different objectives becomes more disorganized [8]. Therefore, it is deficient to only strengthen the discrimination ability of MOEAs elaborately in the objective space. It is also necessary to make the search policies adopted by the generator scalable to cope with the challenges brought by the "*curse of dimensionality*" in the search space. However, the canonical genetic operators (e.g.,

Songbai Liu, Qiuzhen Lin, and Jianqiang Li are with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China (email: songbai@szu.edu.cn).

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

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

Scalable aspects	Scaling-up MOPs	Main modules	Basic strategies	Challenges or limitations
S1 − ϒ: objective space	✡ many-objective MOPs	✓ Discriminator; *//outputs survival solutions by filtering poorly performed solutions from the inputted population*	Environmental selection strategies: ⌖ Pareto-based selection [1]; ⌖ indicator-based selection [2]; ⌖ decomposition-based selection [3]; ⌖ their hybridizations [4];	✧ the insufficient selection pressure for Pareto-based strategies as almost all solutions are mutually non-dominated [5]; ✧ the efficacy of decomposition-based strategies deteriorates severely because their pre-specified weight vectors cannot match well with the increasingly complex and irregular Pareto fronts [6]; ✧ the cost of computing performance indicators (e.g., hypervolume) grows exponentially as the number of objectives increases [7].
S2 − Ω: search space	✡ large-scale MOPs	✓ Generator; *//outputs offspring by finding promising solutions in search space based on the inputted parents*	Reproduction or search strategies: ⌖ polynomial mutation; ⌖ simulated binary crossover; ⌖ particle swarm optimization; ⌖ differential evolution; ⌖ evolutionary strategy [9]; ⌖ etc.	✧ search space expands exponentially with the increase of variables, which requires the generator with growing search capabilities [10]; ✧ the linkage and interaction between variables become more intricate; ✧ the contribution of different variables becomes more imbalanced; ✧ the overlap between shared variables that define different objectives becomes more disorganized [8]; ✧ the overly-increasing number of variables will inevitably make function evaluation expensive [11].
S3 − Ψ: mapping diagram	✡ multimodal MOPs ✡ bi-level MOPs	✓ Discriminator; ✓ Generator.	Integrated strategies: ⌖ Integrate diversity strategies cover objective and search space; ⌖ Integrate search strategies cover low-level and up-level variables.	✧ it is challengeable to maintain the population's diversity in objective space and search space simultaneously for multimodal cases; ✧ It is extremely difficult to find the Pareto solutions with both optimal upper-level variables and lower-level variables.
S4 − Γ: computational cost for function evaluation	✡ expensive MOPs	✓ Evaluator; *//outputs objective function values or fitness values for each inputted solution*	Function evaluation strategies: ⌖ evaluate objective values directly from the given expression; ⌖ predict objective values using a surrogate learned from available simulation data when objective functions are unexpressed or noisy.	✧ require prohibitive computational cost (e.g., central processing unit runtimes) for direct function evaluations [56]; ✧ the construction of a surrogate model (with high quality) requires many promising solutions to approximate the properties [15]; ✧ simulation data is often imbalanced with unknown quality in unexpressed or noisy cases; ✧ the approximation errors of learning models increase accordingly, which may heavily mislead the evolutionary search and selection.
S5 − Λ: problem domain	✡ multi-task MOPs ✡ many-task MOPs ✡ sequential MOPs ✡ dynamic MOPs	✓ Discriminator; ✓ Generator; ✓ Evaluator.	Transfer optimization strategies: ⌖ without knowledge transfer by solving each MOP independently; ⌖ implicit transfer via assortative mating and selective imitation [12]-[14];	✧ start from scratch when meeting with new coming MOPs regardless of the related valuable optimization experience; ✧ the generalization ability of traditional MOEAs is so poor that they cannot leverage helpful experience to achieve more promising optimization efficacy; ✧ the implicit transfer paradigm with random and blind knowledge transfer fails to improve the overall optimization performance.

simulated binary crossover plus polynomial mutation, particle swarm optimization, differential evolution, evolutionary strategy [9], or their hybridizations) used in most MOEAs have been proved to be ineffective in handling the large-scale search space [10]. Furthermore, the overly-increasing number of variables will inevitably make function evaluation correspondingly expensive, such as in high-fidelity computer simulation and super network architecture search [11].

Thirdly, traditional MOEAs focus on solving each MOP independently, so they must start from scratch when meeting with new coming MOPs regardless of the related valuable optimization experience. Under this optimization paradigm, the generalization ability of these MOEAs is so poor that they cannot leverage helpful experience to achieve more promising optimization efficacy when they are challenged by multiple different MOPs sequentially or simultaneously. Considering the optimization of an MOP as a task, the principal goal of solving SMOPs, MMOPs, or DMOPs is to adaptively exploit the complementarities among all the tasks, helping the target task to achieve the most efficient optimization through the transfer of valuable knowledge. Here, the DMOP change to a new environment can be regarded as a new task. Recently, a multiobjective multifactorial EA (MOMFEA) [12] and its subsequent improved variants, e.g., MOMFEA-II [13], and MOMFEA-AKT [14], are proposed to solve MMOPs by exchanging the search experiences between different tasks implicitly. In this multifactorial optimization paradigm, the generator allows, with a certain probability, the solutions from different tasks to exchange information during reproduction via assortative mating and selective imitation [12]. However, when scaling up to the many-task scenario, this MFEA paradigm based on random and blind knowledge transfer fails to improve the overall optimization performance [15].

As discussed above, traditional MOEAs face great challenges in responding to the ever-growing dimensionality and complexity of MOPs, which spring up exuberantly in practical applications from deep learning models with ever-expanding architecture to engineering systems with ever-complex linkage. Thus, to continue being applicable to realistic MOPs, MOEAs should learn to cope with new challenges brought by these complex MOPs. Intuitively, MOEAs can improve their generalization ability to solve scaling-up MOPs by equipping them with ML techniques since the problem-specific characteristics, if well learned, can be a great help for enhancing their adaptive ability of search and selection [16].

2. Discusses on learnable discriminators via clustering

Different from the fixed clustering-guided strategies, these dynamic clustering methods need to iteratively run many times to obtain multiple final clusters with more balanceable diversity and convergence. Nevertheless, there are three key concerns to be noted in designing dynamic clustering-guided strategies:

(1) *How to select a suitable clustering method*? Many angle-based selection strategies can be seen as a variant process of hierarchical clustering. Although hierarchical clustering is commonly used to guide environmental selection, few efforts explain why this method is the most appropriate and whether this method is suitable for clustering the population in the objective space.

(2) *How to select the similarity metric*? Using Euclidean distance for clustering may lead to some dominated solutions (or dominance-resistant solutions) far away from the PF being divided into a separate cluster and having a high probability of surviving to the next generation, which will prevent the population from moving towards PF. To address this issue, cosine similarity is widely used for clustering the population. Nevertheless, the fairness of cosine similarity defined based on the ideal point is still dependent on the curvature of the target PF, so it is often necessary to predict the curvature of PF in advance and then define the corresponding cosine similarity.

(3) *How to select a representative solution in each cluster*? Generally, the solution with the best convergence is chosen as the representative of each cluster, so properly evaluating the convergence of solutions clustered in a local constrained space becomes very critical. The weighted sum indicator is widely used to reflect the convergence of solutions in solving MaOPs.

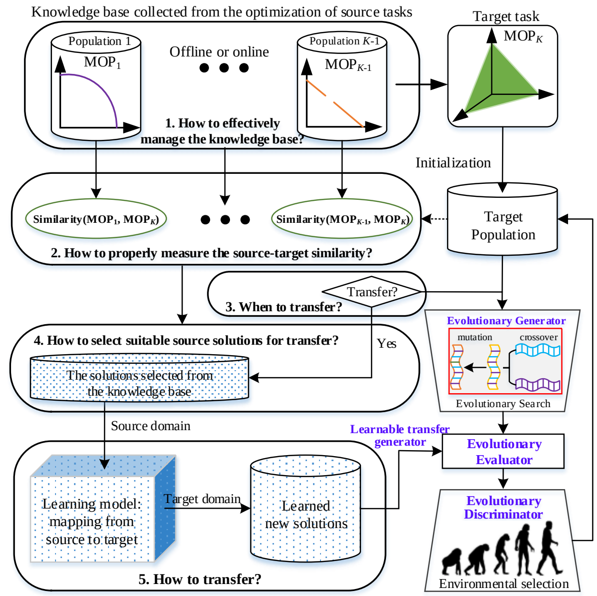

Fig. 6 The basic framework of solution-based transfer generators and the main challenges when designing this kind of transfer generators.

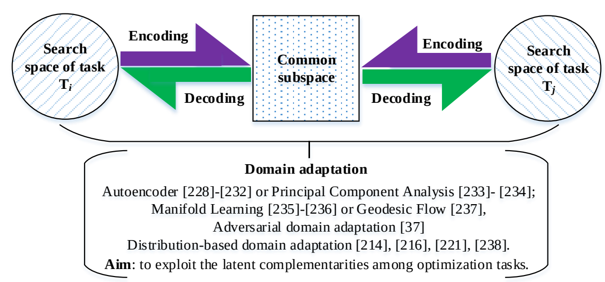

Fig. 7 The basic process and basic idea of a domain adaptation based on common subspace learning.

References

[1] Y. Jin, et al., "Pareto-Based Multiobjective Machine Learning: An Overview and Case Studies," *IEEE trans. syst. man cybern., Part C Appl. rev*., vol. 38, no. 3, pp. 397-415, 2008.

[2] K. Shang, et al., "A Survey on the Hypervolume Indicator in Evolutionary Multiobjective Optimization," *IEEE Trans. Evol. Comput*., vol. 25, no. 1, pp. 1-20, 2021.

[3] K. Li, "Decomposition Multi-Objective Evolutionary Optimization: From State-of-the-Art to Future Opportunities," arXiv:2108.09588v1, 2021.

[4] K. Sindhya, K. Miettinen, K. Deb, A Hybrid Framework for Evolutionary Multi-objective Optimization, *IEEE Trans. Evol. Comput*., vol. 17, no. 4, 495-511, 2013.

[5] K. Deb, H. Jain, "An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints," *IEEE Trans. Evol. Comput*., vol. 18, no. 4, pp. 577-601, 2014.

[6] H. Ishibuchi, et al., "Performance of Decomposition-Based Many-Objective Algorithms Strongly Depends on Pareto Front Shapes," *IEEE Trans. Evol. Comput*., vol. 21, no. 2, pp. 169-190, 2017.

[7] J. Deng, Q. Zhang, "Approximating Hypervolume and Hypervolume Contributions Using Polar Coordinate," *IEEE Trans. Evol. Comput*., vol. 23, no. 5, pp. 913-918, 2019.

[8] S. Z. Martínez, et al., "A Review of Features and Limitations of Existing Scalable Multiobjective Test Suites," *IEEE Trans. Evol. Comput*., vol. 23, no. 1, pp. 130-142, 2019.

[9] K. Zhang, et al., "Evolution Strategy-Based Many-Objective Evolutionary Algorithm Through Vector Equilibrium," *IEEE Trans., Cybern*., vol. 51, no. 11, pp. 5455-5467, 2021.

[10] S. Liu, et al., "Evolutionary Large-Scale Multiobjective Optimization: Benchmarks and Algorithms," *IEEE Trans. Evol. Comput*., in press, 2021.

[11] H. Wang, Y. Jin, J. Doherty, "A Generic Test Suite for Evolutionary Multifidelity Optimization," *IEEE Trans. Evol. Comput*., vol. 22, no. 6, pp. 836-850, 2018.

[12] A. Gupta, et al., "Multiobjective Multifactorial Optimization in Evolutionary Multitasking," *IEEE Trans. Cybern*., vol. 47, no. 7, pp. 1652-1665, 2017.

[13] K. K. Bali, et al., "Cognizant Multitasking in Multiobjective Multifactorial Evolution: MO-MFEA-II," *IEEE Trans. Cybern*., vol. 51, no. 4, pp. 1784-1796, 2021.

[14] L. Zhou, et al., "Toward Adaptive Knowledge Transfer in Multifactorial Evolutionary Computation," *IEEE Trans. Cybern*., vol. 51, no. 5, pp. 2563-2576, 2021.

[15] K. Deb, et al., "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II," *IEEE Trans. Evol. Comput*., vol. 6, no. 2, pp. 182-197, 2002.

[16] W. Li, G. Wang, A. H. Gandomi, "A Survey of Learning-Based Intelligent Optimization Algorithms," *Arch. Comput. Methods Eng*., vol. 28, pp. 3781-3799, 2021.

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

Note: The references/citations in this table correspond to the references in the published paper.