Artificial Intelligence Review
https://doi.org/10.1007/s10462-021-10042-y

A survey on evolutionary computation for complex continuous optimization

Zhi-Hui Zhan[1,2] · **Lin Shi**[1,2] · **Kay Chen Tan**[3] · **Jun Zhang**[4,5]

Accepted: 9 July 2021

Abstract

Complex continuous optimization problems widely exist nowadays due to the fast development of the economy and society. Moreover, the technologies like Internet of things, cloud computing, and big data also make optimization problems with more challenges including *M*any-dimensions, *M*any-changes, *M*any-optima, *M*any-constraints, and *M*any-costs. We term these as 5-M challenges that exist in large-scale optimization problems, dynamic optimization problems, multi-modal optimization problems, multi-objective optimization problems, many-objective optimization problems, constrained optimization problems, and expensive optimization problems in practical applications. The evolutionary computation (EC) algorithms are a kind of promising global optimization tools that have not only been widely applied for solving traditional optimization problems, but also have emerged booming research for solving the above-mentioned complex continuous optimization problems in recent years. In order to show how EC algorithms are promising and efficient in dealing with the 5-M complex challenges, this paper presents a comprehensive survey by proposing a novel taxonomy according to the function of the approaches, including *reducing problem difficulty*, *increasing algorithm diversity*, *accelerating convergence speed*, *reducing running time*, and *extending application field*. Moreover, some future research directions on using EC algorithms to solve complex continuous optimization problems are proposed and discussed. We believe that such a survey can draw attention, raise discussions, and inspire new ideas of EC research into complex continuous optimization problems and real-world applications.

Keywords Evolutionary computation (EC) · Evolutionary algorithm (EA) · Swarm intelligence (SI) · Complex continuous optimization problems · Large-scale optimization · Dynamic optimization · Multi-modal optimization · Many-objective optimization · Constrained optimization · Expensive optimization · Function-oriented taxonomy

✉ Zhi-Hui Zhan
zhanapollo@163.com

Extended author information available on the last page of the article

Published online: 27 July 2021

1 Introduction

Optimization is frequently encountered in many fields. People could use try-and-error methods to test different solutions for very simple optimization problems. However, with the development of the economy and society, problems become more complex that the try-and-error methods are no longer suitable. Therefore, many mathematical-based and computer-aid optimization techniques have been developed. Among them, evolutionary computation (EC) (Fogel 1995) has become a good kind of global optimization technique in many optimization problems. The EC technology dates from the 1960s when evolutionary algorithms (EA) like genetic algorithm (GA), evolutionary programming (EP), evolution strategies (ES), and genetic programming (GP) were proposed for solving global optimization problems (Eiben and Smith 2015). The EA simulates the evolutionary process of biology and the natural selection principle to optimize the problems. Some optimization techniques like differential evolution (DE) and estimation distribution of algorithm (EDA) that appeared from the 1990s are also regarded as EA. Besides the EA, some other optimization techniques also appeared since the 1990s that simulated the intelligent behaviors of swarm such as ants and birds. They include ant colony optimization (ACO) and particle swarm optimization (PSO), which are also known as swarm intelligence (SI) algorithms (Bonabeau et al. 2000). Nowadays, the EC family mainly refers to both the EA and SI, and has developed fast in recent three decades since the 1990s (Zhang et al. 2011).

Although EC algorithms have been successfully applied to many kinds of global optimization problems, the new difficulties in optimization problems have also brought great challenges to EC algorithms in the recent decades. As we know, due to the development of the Internet of things, cloud computing, and big data, optimization problems nowadays have become more and more complex. For example, in the big data environments, the complex optimization problems, like many other big data problems, always have the so-called 4-V challenges as *V*olume, *V*elocity, *V*ariety, and *V*alue (Yin and Kaynak 2015), which respectively mean the amount of data, the speed of change, the range of data, and the validity of data. Specifically, these complex optimization problems are usually large-scale, dynamic, with many local/global optima, with constraints, with many objectives, and with very expensive objective function evaluation. In this paper, we the first time introduce a 5-M concept to classify the complex continuous optimization problems into 5-M categories, including *M*any-dimensions, *M*any-changes, *M*any-optima, *M*any-constraints, and *M*any-costs. Therefore, the complex continuous optimization problems typically include optimization problems known as large-scale optimization problems (LSOP), dynamic optimization problems (DOP), multimodal optimization problems (MMOP), multi-objective optimization problems (MOP), many-objective optimization problems (MaOP), constrained optimization problems (COP), and expensive optimization problems (EOP) in the EC community. These above-mentioned complex continuous optimization problems are corresponding related to the 5-M and 4-V challenges as shown in Fig. 1.

When dealing with complex continuous optimization problems, the traditional EC algorithms may have great potential but also face great challenges. Therefore, research into EC algorithms in solving complex continuous optimization problems has become a new trend in EC in the recent decades and a lot of works have been proposed. However, the related works are scattered in the literature and few systematic surveys can be found. To fill this gap, this paper aims to provide a comprehensive and concrete survey on these scatted works by proposing a systematic and structural function-oriented taxonomy to fully review and analyze how to enable and enhance EC algorithms in efficiently solving

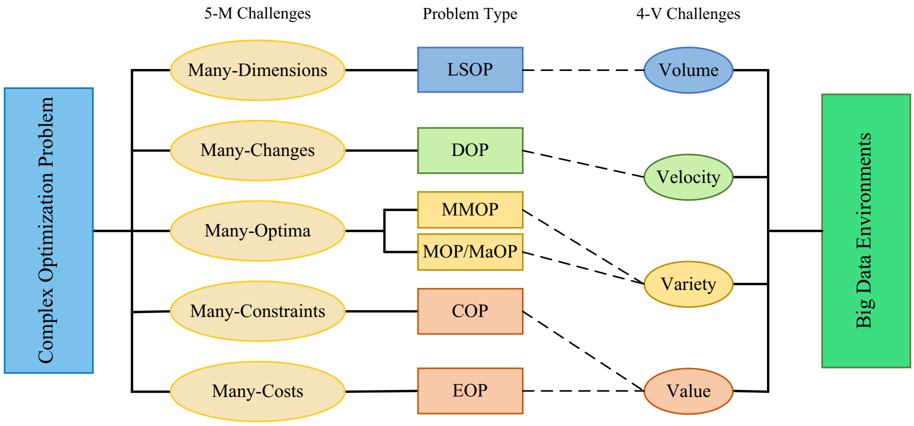

Fig. 1 The relationship of complex continuous optimization problems with 5-M and 4-V challenges

complex continuous optimization problems. To concentrate on this topic, we mainly focus on research works of using EC algorithms for solving complex continuous optimization problems in this survey. Moreover, in order to better illustrate the most representative approaches for solving complex continuous problems, we prefer to cite works by comprehensively considering (but not limited to) their sources (e.g., the reputed journals and conferences in the EC community), publication years (e.g., mainly in recent 5 years), and impact (e.g., the recognition and citation). The main contributions of this paper are summarized as follows.

Firstly, the concept of the 5-M challenges is introduced to describe the difficulty of complex continuous optimization problems. This is consistent with the 4-V challenges in big data complex environments and is helpful to understand why and how the complex continuous optimization problems are difficult and challenging.

Secondly, a function-oriented taxonomy is introduced to systematically and structurally classify the existing works according to their functions on how to enable and enhance the EC algorithms to solve complex continuous optimization problems efficiently.

Thirdly, some future research directions and open problems on using EC algorithms to solve complex continuous optimization problems are proposed and discussed. This will certainly encourage and promote research works in this field to be wider and deeper.

The rest of this paper is organized as follows. Section 2 discusses the main challenges in different kinds of complex continuous optimization problems and introduces the function-oriented taxonomy. Section 3 surveys research works according to the taxonomy on accommodating EC algorithms in solving complex continuous optimization problems. In Sect. 4, some future research directions are proposed and discussed, followed by the conclusions in Sect. 5.

2 Challenges and taxonomy

Due to the 5-M challenges, complex continuous optimization problems are more difficult than traditional continuous optimization problems. Specifically, the challenges in the LSOP, DOP, MMOP, MOP/MaOP, COP, and EOP are summarized in Table 1.

Table 1 The challenges in different types of complex continuous optimization problems

5-M	Problem Type	Challenges
*M*any dimensions	LSOP	Dimensions are very large, always more than 1000 How to avoid local optima in curse of dimensionality? How to converge in the very huge search space?
*M*any changes	DOP	Environment changes frequently during the time How to jump out of previous optimum? How to fast find the new optimum?
*M*any optima	MMOP	A set of global optima in different regions How to find all of the global optima? How to refine the accuracy of each global optimum?
	MOP/MaOP	A set of Pareto optimal solutions on the Pareto front How to better converge to the global Pareto front? How to maintain diversity along the whole Pareto front?
*M*any constraints	COP	Solutions easily become infeasible due to the constraints How to deal with the infeasible solution? How to approach the feasible region to find the optimum?
*M*any costs	EOP	The computational burden is expensive/time-costing How to reduce the optimization time? How to increase the solution accuracy in limited time?

Due to these challenges and difficulties, we have to accommodate the EC algorithms to fit the complex problems. The approaches can be considered from three angles. Firstly, as the problem is complex, can we reduce the difficulty of the original problem so that the EC algorithms can solve it? For example, decomposing or transforming the original complex problems into other simple problems. This can be regarded as the "*reducing problem difficulty*" approach. Secondly, as the EC algorithms may deteriorate their performance in complex search environments, can we enhance the ability of the algorithms? For example, if the algorithm is easy to fall into local optima, we can increase diversity to make the algorithm with stronger global search ability. Generally speaking, the diversity of the algorithm is related to the population information, such as population position data and population movement data, and therefore can be enhanced by using various parameters/operators and multiple populations. For another example, if the search space is large or the computation is time-costing, we can accelerate the convergence speed or reduce the running time. These can be regarded as the "*increasing algorithm diversity*," "*accelerating convergence speed*," and "*reducing running time*" approaches. Herein, the larger diversity is generally related to stronger exploration ability, while faster convergence speed is generally related to better exploitation ability. Thirdly, as many complex continuous optimization problems come from real-world applications, can we make research into dealing with these problems by considering their practical characteristics? This can be regarded as the "*extending application field*" approach.

According to the preceding discussed functions of the approaches, this paper proposes a function-oriented taxonomy to systematically and structurally classify research works on EC algorithms for solving complex continuous optimization problems, as shown in Fig. 2.

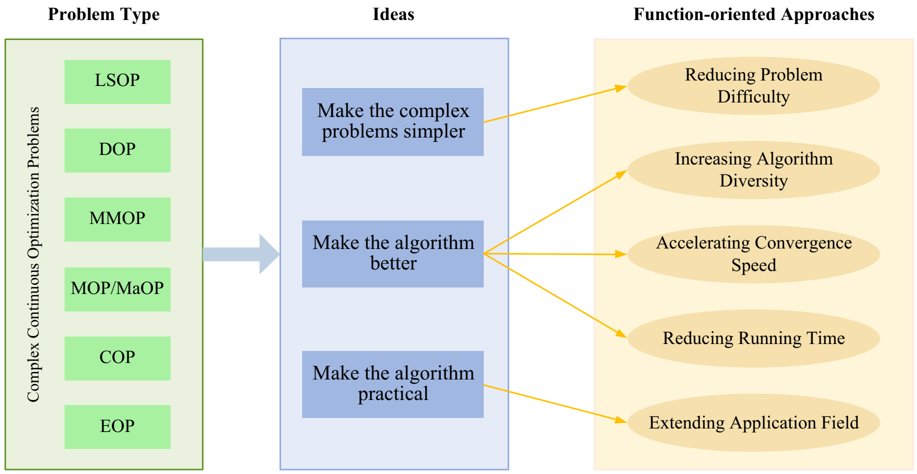
Problem Type
Ideas
Function-oriented Approaches
Complex Continuous Optimization Problems
LSOP
DOP
MMOP
MOP/MaOP
COP
EOP
Make the complex problems simpler
Make the algorithm better
Make the algorithm practical
Reducing Problem Difficulty
Increasing Algorithm Diversity
Accelerating Convergence Speed
Reducing Running Time
Extending Application Field

Fig. 2 Illustration of the taxonomy based on the function-oriented approaches

3 EC algorithms for complex continuous optimization problems

This section reviews the EC algorithms for solving complex continuous optimization problems according to the taxonomy in Fig. 2. Specifically, the survey is classified into six parts according to the problem type, i.e., EC algorithms for LSOP, DOP, MMOP, MOP/MaOP, COP, and EOP, respectively. In each part, the related works are classified according to our function-oriented taxonomy, including some or all of the *reducing problem difficulty*, *increasing algorithm diversity*, *accelerating convergence speed*, *reducing running time*, and *extending application field* approaches.

3.1 EC for large-scale optimization problems

Large-scale optimization problems (LSOP) are very common in many research fields of science and engineering. The number of decision variables (namely dimensions) of LSOP is relatively large, which is generally more than 500 and usually more than 1000. As the dimension increases, LSOP becomes more and more difficult to be solved. On the one hand, the search space of LSOP will increase exponentially and on the other hand, the number of local optimal solutions may also increase exponentially. As a result, the performance of most traditional EC algorithms will deteriorate rapidly when solving LSOP because they easily fall into the local optima (Jian et al. 2020). Therefore, we need to adopt some appropriate approaches according to the characteristics of LSOP to improve the performance of EC algorithms in solving LSOP.

In some existing literature (Jian et al. 2020), the approaches of EC algorithms to solve LSOP are mainly divided into two categories: one is the cooperative co-evolution (CC) method which decomposes the whole LSOP into several subproblems, and the other is the non-CC method which adopts some additional strategies to enhance the algorithm performance, as shown in the left part of Fig. 3. In order to survey the EC algorithms for LSOP in a more detailed and functional-oriented angle, we classify research works into three categories according to our function-oriented taxonomy, as shown in the right part of Fig. 3. The first is *reducing problem difficulty*, mainly by adopting the CC method to

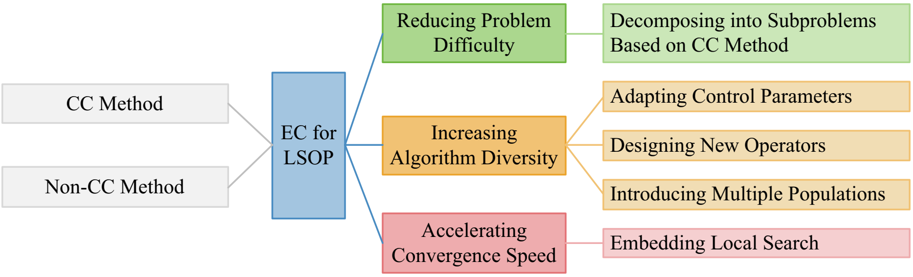
CC Method
Non-CC Method
EC for LSOP
Reducing Problem Difficulty
Increasing Algorithm Diversity
Accelerating Convergence Speed
Decomposing into Subproblems Based on CC Method
Adapting Control Parameters
Designing New Operators
Introducing Multiple Populations
Embedding Local Search

Fig. 3 Approaches of EC algorithms for solving LSOP

decompose the entire LSOP into several subproblems which are relatively small scale. The second is *increasing algorithm diversity* so that the algorithm can search for more high-accuracy solutions in the large search space, including adapting control parameters, designing new operators, and introducing multiple populations. The third is *accelerating convergence speed*, mainly by embedding local search, so as to make the algorithms applicable to find promising solutions in an acceptable time. In addition, some application-oriented approaches for LSOP will be presented at the end.

3.1.1 Reducing problem difficulty

The CC method which exploits the idea of "divide-and-conquer" is a famous and common method to reduce LSOP difficulty. In other words, the CC method aims to decompose an entire LSOP into several subproblems that have fewer decision variables and are easier to be solved. The CC method was firstly proposed by Potter and De Jong (1994) in GA, termed as CCGA, which decomposed a D-dimensional LSOP into D 1-dimensional subproblems. Similarly, Liu et al. (2001) adopted this method to fast EP (FEP), forming FEPCC. However, CCGA and FEPCC perform poorly when solving variables nonseparable functions. Later, van den Bergh and Engelbrecht (2004) modified this decomposition strategy and applied it to the PSO, named CPSO-S$_K$. Unlike CCGA and FEPCC, CPSO-S$_K$ decomposes a D-dimensional problem into k s-dimensional subproblems ($s \ll D$, $s = D/k$), that each subproblem contains s decision variables. In order to further improve the performance of CPSO, Li and Yao proposed two improved variants named CCPSO (Li and Yao 2009) and CCPSO2 (Li and Yao 2012) that adopted the random grouping technique proposed by Yang et al. (2008a) for variables decomposition. Most recently, Zhang et al. (2020c) proposed to use function independent decomposition strategy to decompose the LSOP. In the DE algorithm, Shi et al. (2005) first applied the CC method to DE and proposed a new decomposition strategy, called splitting-in-half strategy, which decomposed all the decision variables into two equal subcomponents. In order to further reduce the scale of subproblems, Yang et al. (2008a) proposed a new algorithm called DECC-G (where G means grouping) which used a random grouping (RG) strategy. Besides, Yang et al. (2008b) proposed a multilevel CC (MLCC) method to ease the burden of setting the group size. In MLCC, a set of possible group sizes are provided instead of using a fixed value. Later, several DE-based decomposition strategies which can detect the relationship between decision variables (interacted decision variables will be assigned to the same subproblem) have been proposed, such as delta grouping strategy (DECC-D, DECC-DML) (Omidvar et al. 2010), differential grouping (DG) strategy (DECC-DG) (Omidvar et al.

2014), extended DG strategy (DECC-XDG) (Sun et al. 2015), graph-based DG strategy (DECC-gDG) (Ling et al. 2016), improved variant of the DECC-DG (DECC-DG2) (Omidvar et al. (2017), recursive DG (RDG) strategy (DECC-RDG) (Sun et al. 2018a), DG with spectral clustering strategy (DGSC) (Li et al. 2019a), and soft grouping strategy (SGCC) (Liu et al. 2019b), which can increase the grouping accuracy better and better. Nowadays, evolution strategies (ES) are getting popular in solving complex problems (Müller and Glasmachers 2018). The CC method is also applied in the covariance matrix adaption ES (CMA-ES) as the CC-CMA-ES (Liu and Tang 2013). Later, Mei et al. (2016) proposed a global DG strategy (GDG), which could improve decomposition accuracy by maintaining the global information, and also applied it to CMA-ES, forming CC-GDG-CMA-ES. Inspired by the DG2 and RDG strategy, Sun et al. proposed two RDG variants which combined with CMA-ES, called RDG2 (Sun et al. 2018b) and RDG3 (Sun et al. 2019a), respectively, to better discover the relationship between decision variables and decompose the overlapping problems. In addition, RDG3 is the winning algorithm in the IEEE Congress on Evolutionary Computation (CEC) 2019 Competition. Moreover, Peng et al. (2019) introduced a multimodal optimization strategy into the CC method (MMO-CC) to optimize subproblems after decomposing the LSOP, which could reduce the consumption of fitness evaluations (FEs). The development roadmap of the EC algorithms based on the CC method is illustrated in Fig. 4, showing that the research into using the CC method to reduce problem difficulty for efficiently solving LSOP has a long-term development and is still very active.

The CC-based algorithms mentioned above treat the subproblems equally. That is, all subproblems are optimized by the same number of computational resources. However, as the number of variables in different subproblems is different, there is an imbalance among subproblems. Therefore, different computational resources should be assigned to different subproblems. For example, Omidvar et al. (2011) proposed two contribution-based CC methods for LSOP, called CBCC1 and CBCC2. These two methods alleviate the imbalance of the subproblems to use the computational resources more efficiently. Later, in order to find a better balance between exploration and exploitation, Omidvar et al. (2016) proposed a new contribution-based algorithm called CBCC3. Besides, Yang et al. (2017a) proposed

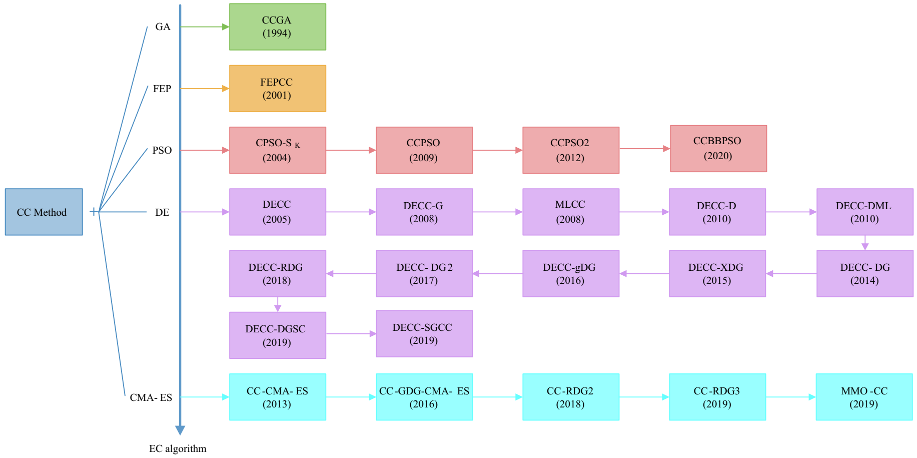

Fig. 4 The development roadmap of the algorithms based on CC method

a new CC framework based on the contribution of the subpopulation. The new CC framework can not only save computational resources by checking whether a subpopulation is stagnant but also update the contribution of a subpopulation dynamically. In addition, some adaptive computation resource allocation strategies combined with CC methods have been proposed, including boosting CC method (Ren et al. 2019), dynamic CC method (Zhang et al. 2019c), and distributed CC method (Sun et al. 2019a). Recently, Irawan et al. (2020) proposed a two-stage CC method with CMA-ES to solve LSOP. This algorithm is devoted to decision variable decomposition and efficient resource allocation in these two stages, respectively. Moreover, a Bandit-based CC method that can take account of the imbalance of subproblems in LSOP has been proposed by Kazimipour et al. (2019).

3.1.2 Increasing algorithm diversity

Due to the insufficient search diversity caused by the high dimension of LSOP, some algorithms may get trapped in the local optima. Therefore, we can improve the performance of algorithms in solving LSOP by increasing algorithm diversity. The approaches can be mainly divided into three categories: *adapting control parameters*, *designing new operators*, and *introducing multiple populations*.

(a) Adapting control parameters
Some parameters in EC algorithms that may influence algorithm diversity can be adaptively controlled. Takahama and Sakai (2012b) proposed a DE with landscape modality detection, called LMDEa, for LSOP that could self-adapt the control parameters dynamically by modality detection. The LMDEa also uses a diversity archive to store defeated trial vectors to keep diversity. Later, Kushida et al. (2015) improved the LMDEa by introducing the concept of rank-based DE (Takahama and Sakai 2012a), resulting in LMRDEa. The LMRDEa considers the diversity-convergence balance better and controls the parameters as well as the mutation strategy by detecting the landscape modality. The jDE (Brest et al. 2006) and JADE (Zhang and Sanderson 2009) are also two well-known DE variants that featured with adapting control parameters strategies. Many algorithms based on these two algorithms for LSOP have been proposed, including jDE with small and varying population size (Brest et al. 2012), jDE with a population size reduction mechanism (Brest and Maucec 2011), and JADE with sorting crossover rate (Zhou et al. 2017). Recently, Zhang et al. (2020c) proposed a dual control strategy to self-adapt the population size in DE, called APDE, so as to increase the population size to maintain diversity.

(b) Designing new operators
In recent years, many researchers have also designed new operators for EC algorithms to improve their performance in solving LSOP. For example, many researchers modify the velocity and position update methods of the standard PSO to increase algorithm diversity. Cheng and Jin proposed two PSO variants to solve LSOP. One is a competitive swarm optimizer (CSO) (Cheng and Jin 2015a) and the other is a social learning PSO (SLPSO) (Cheng and Jin 2015b). In CSO (Cheng and Jin 2015a), a pairwise competition mechanism is introduced and the losing particle will learn from the winner particle to increase the diversity of the population. In SLPSO (Cheng and Jin 2015b), all particles in the population are sorted before the evolutionary process, and each particle learns from any particle that is better than itself. A segment-based predominant learning swarm optimizer (SPLSO) (Yang et al. 2017b) and a level-based

learning swarm optimizer (Yang et al. 2018) were proposed by Yang et al. for solving LSOP, where the predominant guidance strategy and two-level guidance strategy were respectively proposed to increase the population diversity. In SPLSO, all dimensions of each particle are randomly divided into several segments, and the dimension of each segment is updated by learning from different predominant particles, which can further increase population diversity. In addition, Guo et al. (2018) proposed to only use the personal historical best position (***pbest***) instead of both ***pbest*** and the global best position (***gbest***) like standard PSO to guide particle updating, so as to avoid the algorithm falling into the local optima prematurely. Inspired by CSO and SLPSO, Lan et al. (2020) and Deng et al. (2019) proposed a two-phase learning based swarm optimizer and a ranking-based biased learning swarm optimizer to solve LSOP, respectively, where both need to group the particles in the population, and then the worse particle will learn from the better particle. Similarly, the mutation operator in standard DE is also modified to increase population diversity. Zhao et al. (2011) incorporated the mutation strategy in JADE (Zhang and Sanderson 2009) and modified multi-trajectory search (MMTS) into the self-adaptive DE, resulting in the SaDE-MMTS algorithm. Ao (2012) proposed an enhanced DE for LSOP called SMDE that employed two mutation variants to increase algorithm diversity. Moreover, a multiple parents guided DE (Yang et al. 2016) was proposed by Yang et al. for LSOP, where the multiple guidance strategy was proposed to improve the performance in solving LSOP. Furthermore, LaTorre et al. (2012) developed a multiple offspring sampling framework, which hybridized different algorithms to deal with LSOP.

(c) Introducing multiple populations
Introducing multiple populations is also a strategy to increase algorithm diversity. Weber et al. (2011) proposed a shuffle or update parallel DE (SOUPDE) which was a multi-populations algorithm. The SOUPDE adopts a shuffling operation by randomly rearranging the individuals over the subpopulations to avoid premature convergence. Similarly, Wang et al. (2013a) proposed a parallel DE (PDE) with multiple populations based on graphics processing units called GOjDE for LSOP. Ge et al. (2016) proposed an individual migration strategy based on the diversity of subpopulations and combined it with the multi-populations DE algorithm to solve LSOP. Li et al. (2021) proposed a multi-population PSO variant for LSOP, which could self-adapt the subpopulation size and balance the exploration and exploitation abilities.

In addition, many distributed DE (DDE) variants and distributed PSO (DPSO) variants have been proposed for solving LSOP. Weber et al. (2009) proposed a DDE that divided the whole population into two subpopulations, namely DDE with explorative-exploitative population families. Cheng et al. (2013) proposed two migration selection strategies, target individual based migration selection strategy, and representative individual based migration selection strategy, to increase the diversity of the DDE. Ge et al. (2018b) designed novel mergence and split operators which could make full use of population resources, and applied it to DDE, forming DDE-AMS. Later, he proposed another DDE variant called competition-based distributed DE (DDE-CB) (Ge et al. 2018a) to solve LSOP, where two operators named opposition-invasion and cross-invasion were used to dynamically change the structure of subpopulations. In the DPSO research, Yang et al. (2020) proposed a distributed swarm optimizer based on a master–slave model to solve LSOP, where the asynchronous and adaptive communication between the master and each slave is adaptively triggered according to the search state of the associated swarm. Wang et al. (2020c) introduced a dynamic group learning strategy (DGL) into DPSO so that different subpopulations could cooperate with each other to further increase the algorithm diversity. Moreover, the authors also further designed an adaptive strategy to control the size of the subpopulation, resulting in adaptive granularity learning (AGL)-based DPSO for LSOP (Wang et al. 2021). These two DPSO variants are named DGLDPSO and AGLDPSO, which are also the very recent and well-performing EC algorithms for solving LSOP.

3.1.3 Accelerating convergence speed

Due to the high dimension of LSOP, there exists a difficulty in convergence speed. So the convergence speed is also an issue that needs to be considered when solving LSOP. Embedding local search is the main method for accelerating the convergence speed. Zhao et al. (2008) introduced the quasi-Newton method to perform local search for accelerating the convergence speed in PSO. Zhang and Chiang (2017) designed a local search strategy and combined it with PSO in solving LSOP. This local search strategy not only can accelerate the convergence speed, but also can find a set of high-quality local optimal solutions or even global optimal solutions. In addition, Molina et al. proposed three large-scale optimization algorithms based on local search, including memetic algorithm based on local search chains (Molina et al. 2010), iterative hybridization of DE with local search (IHDELS) (Molina and Herrera 2015), and SHADE-ILS (Molina et al. 2018) which was the success-history based parameter adaption DE (SHADE) (Tanabe and Fukunaga 2013) with iterative local search. Among them, both IHDELS and SHADE-ILS improve the performance by accelerating the convergence speed in solving LSOP via iteratively embedding different local search strategies. Besides, Xie et al. (2013) proposed a gradient-based local search into DE to accelerate the convergence speed. Moreover, an adaptive population DE with a local search which could enhance the population's solution search ability and accelerate the convergence speed for solving LSOP was proposed by Hsieh et al. (2012). Recently, Yildiz and Topal (2019) designed a directional local search (DLS) strategy to improve the exploitation ability of the algorithm and accelerate the convergence speed. The DLS strategy is embedded in a DE variant and this algorithm achieves promising performance in solving LSOP. Most recently, Jian et al. (2021) proposed a novel region encoding scheme (RES) to help EC algorithms evolve faster in solving LSOP. The RES changes the solution encoding from traditional point-based to region-based, so that a solution can carry out region search to search for better solutions near its current point, offering a greater chance to discover the nearby optimal solutions and helping to accelerate the convergence speed of the whole population.

3.1.4 Extending application field

LSOP is very common in various fields in the real world and many algorithms have been proposed to solve practical large-scale optimization applications in science and engineering. In solving electroencephalogram classification problems, Guerrero-Mosquera et al. (2011) proposed a dimensionality reduction method, which could reduce the size of the feature matrix by using mutual information to improve the performance of the classifier. In solving cloud resources scheduling problems, Zhan et al. (2015) presented a comprehensive survey of cloud computing resource scheduling by EC algorithms. Tan et al. (2020) combined the genetic programming algorithm with the CC method (CCGP) to deal with the online resource allocation problem in container-based cloud environments. CCGP can deal with two interact allocations (containers to virtual machines (VMs) and VMs to physical machines) separately to improve the performance. Wang et al. (2020c) introduced multiple population strategy into PSO to solve large-scale cloud workflow scheduling problems and achieved promising results. Liu et al. (2018b) designed a local search strategy to deal with the virtual machine placement problem in cloud computing, which helped the algorithm find the optima more quickly. In solving capacitated arc routing problems, Mei et al. (2014) adopted the CC framework and proposed an effective decomposition scheme, called the Route Distance Grouping, to decompose the problem. In solving large-scale supply chain network design problems, Zhang et al. (2020b) applied the CC method to barebones PSO to propose a CCBBPSO algorithm for large-scale supply chain network design with uncertainties (LUSCND). In solving community detection and inference problems, Ma et al. (2019) proposed an expectation maximization algorithm, which could reduce the link error and improve the reliability of platforms' observations. In solving large-scale media access control problem, Nekooei and Chen (2020) introduced a multiple population strategy in PSO and achieved promising results. In conclusion, LSOP is an important branch of complex continuous optimization problems and many large-scale real-world problems are attracting increasing attention from researchers. Moreover, EC algorithms have a good advantage in solving LSOP.

3.2 EC for dynamic optimization problems

Many real-world optimization problems are dynamic. Compared with static optimization problems, the environment of dynamic optimization problems (DOP) may change over time, including the change of the location of the optimal solution, the dimension, and the search space of the problem. When solving DOP, the EC algorithms not only need to find the global optimal solution in a specific environment, but also need to track the changing optima in different environments. Thus, the main challenges of solving DOP by EC algorithms are how the population jumps out of previous optimum when the environment changes and how the algorithm finds the new optimal solution as fast as possible in the new environment.

In order to solve DOP more efficiently, many EC algorithms have been proposed. Traditional classification methods of solving DOP are shown in the left part of Fig. 5. Compared with traditional classification methods, we classify the solving approaches according to our function-oriented taxonomy and realize a more comprehensive classification. Therefore, the works can be classified into the following three parts according to the approach functions as: *reducing problem difficulty*, *increasing algorithm diversity*, and

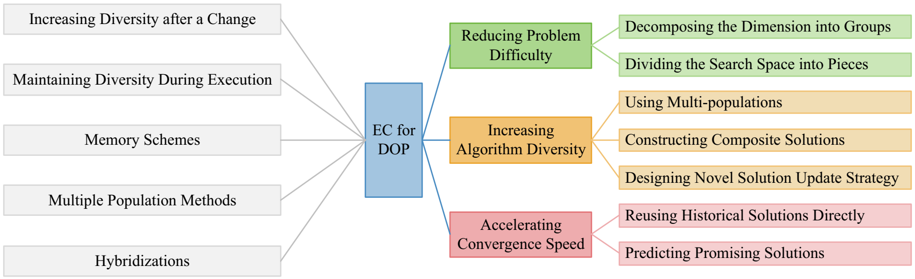
Increasing Diversity after a Change
Maintaining Diversity During Execution
Memory Schemes
Multiple Population Methods
Hybridizations
EC for DOP
Reducing Problem Difficulty
Increasing Algorithm Diversity
Accelerating Convergence Speed
Decomposing the Dimension into Groups
Dividing the Search Space into Pieces
Using Multi-populations
Constructing Composite Solutions
Designing Novel Solution Update Strategy
Reusing Historical Solutions Directly
Predicting Promising Solutions

Fig. 5 Approaches of EC algorithms for solving DOP

accelerating convergence speed, as shown in the right part of Fig. 5. Furthermore, the solving approaches for real-world DOP are surveyed in the *extending application field* part.

3.2.1 Reducing problem difficulty

Considering the difficulty of solving DOP, there is a simple idea that we can reduce the difficulty of problems by dividing the complex DOP into a set of simpler problems. From the perspective of the decision space of the problem, there are two main ways: *decomposing the dimension into groups* and *dividing the search space into pieces*.

The idea of decomposing the dimension into groups comes from the CC method, which has been widely used to solve LSOP. Rakitianskaia and Engelbrecht (2008) proposed a CC-based cooperative charged PSO. By decomposing dimensions of the search space and evolving different dimensions separately, the algorithm performs a good performance in locating and tracking the changing optima in the changing environment. Similarly, Unger et al. (2013) combined quantum PSO with the CC method to solve DOP.

Dividing the search space into pieces is another way to reduce the difficulty of DOP. Hashemi and Meybodi (2009) introduced cellular automata into PSO to address DOP. The Cellular automata partition the search space, so that each partition corresponds to a cell. In cellular PSO, particles in a cell are guided by their best personal solutions and the best solution found in their neighborhood cells, promoting the search for the optimum in each cell. Inspired by cellular PSO, Noroozi et al. (2011) proposed CellularDE for DOP by using the same cellular automaton, in which individuals in each cell were evolved by DE independently. Sharifi et al. (2012) proposed a two phased cellular PSO, which partitioned the search space and involved two phases during the evolution of individuals in each cell. The cells can split the search space, making it easy for the algorithm to find better solutions after environmental changes.

3.2.2 Increasing algorithm diversity

When solving DOP by EC algorithms, the population will gradually converge during the evolutionary process, so that it is necessary to jump out of the previous local region when the environment changes. Maintaining diversity is conducive for the algorithm to avoiding getting trapped in the local region. Then the algorithm is able to continuously explore the search space and find the new optimal solution. Thus, many methods have been proposed to increase the diversity of algorithms, such as *using multi-populations*, *constructing composite solutions*, and *designing novel solution update strategy*.

(a) Multi-populations
Using multi-populations is a common way for algorithms to increase diversity. Herein, multiple populations are maintained during the evolution and each population takes responsibility for a separate task. The goals of these tasks may be the same or different. Accordingly, there are two multiple populations models named heterogeneous and homogeneous models. In the heterogeneous model, the multiple populations are in different layers or with different configurations or tasks. In the homogeneous model, each population has the same task. Furthermore, the number and size of populations may be fixed or variable, and the search space of each population may have a fixed

Table 2 The features comparisons of some different multi-population methods in solving DOP

Reference	Model		Population Number		Population Size		Search Space	
	homogeneous	heterogeneous	variable	fixed	variable	fixed	fixed size	overlap
Branke et al. (2000)		✓	✓		✓		×	✓
Li and Yang (2008)		✓	✓			✓	✓	×
Li et al. (2019b)		✓		✓	✓		×	✓
Yang and Li (2010)	✓		✓		✓		×	✓
Nickabadi et al. (2012)	✓		✓		✓		×	–
Halder et al. (2013)	✓		✓		✓		×	×
Li et al. (2016b)	✓		✓		✓		×	×
Luo et al. (2019)	✓		✓		✓		×	×
Zhang et al. (2019b)	✓		✓		✓		×	×

size or overlap with each other. Here, we make a comparison about features of some different multi-population methods, shown in Table 2.

Based on the heterogeneous model, Branke et al. (2000) proposed the self-organizing scouts (SOS) algorithm. The SOS includes a parent population and many child populations. The parent population constantly explores and searches for new optima in the entire search space, while child populations generated by splitting from the parent population are used to track optima. Inspired by the scout model of SOS, Li and Yang (2008) proposed a fast multi-swarm optimization (FMSO) algorithm. In FMSO, a parent population is used to detect the promising area, while child populations are used to search the local optima. Recently, a cloud-based heterogeneous DE named Cloudde has been proposed for global optimization by using a set of heterogeneous populations with different algorithm configurations (Zhan et al. 2017). As Cloudde can increase the algorithm diversity by the migration among different populations when the environment changes, the Cloudde-based distributed DE has been successfully applied to solve DOP efficiently (Li et al. 2019b).

Different from the heterogeneous model, populations in the homogeneous model explore the search space and track local optima at the same time. Many algorithms based on the homogeneous model utilize the clustering method to create subpopulations. Yang and Li (2010) proposed a clustering PSO, where the number of subpopulations and search area of each subpopulation were automatically adjusted through a hierarchical clustering method. Later, they extended the hierarchical clustering method with a random immigrant strategy to regain the algorithm diversity if it decreased to a certain level (Li and Yang 2012). Moreover, Nickabadi et al. (2012) proposed a competitive clustering PSO where a novel clustering method combining the objective function value and spatial distribution of the particles was used to create subpopulations without requiring predefined number and radius of subpopulations. Halder et al. (2013) proposed a cluster-based DE algorithm with an external archive to solve DOP, in which the *k*-means cluster method was used to partition the population and the number of subpopulations was updating over time, resulting in the re-clustering of individuals. To better track multiple optima in DOP, Li et al. (2016b) proposed an adaptive multi-population (AMP) framework. AMP creates non-overlapping populations by a single linkage hierarchical clustering method and adopts an adaptive mechanism that learns from algorithm behavior changes through interacting with environments for dynamically adjusting the number of populations. Recently, Luo et al. (2019) proposed a distributed multiple population framework to increase algorithm diversity for solving DOP. Zhang et al. (2019b) proposed a new cluster-based clonal selection algorithm, where a max–min distance cluster method based on the fitness and Euclidean distance was used to partition the population. Vafashoar and Meybodi (2020) proposed a multi-population DE algorithm, which is different from past heterogeneous algorithms. In this proposed algorithm, each population resides in one cell of a cellular learning automaton, which is the combination of a cellular automaton with learning automata. Furthermore, a collection of three mutation schemes is used to induce a different behavior on a population, and an adaptive mechanism based on cellular learning automaton is utilized to control the application of each mutation scheme.

A new method of using multiple populations to solve DOP is allocating limited computational resources to different populations, which can further improve the effectiveness of the algorithm. Kordestani et al. (2019) presented a novel framework and two methods for scheduling the populations. The first method allocates function evaluations according to the quality of populations and the degree of diversity among them, and the second method uses the learning automata as the central unit for performing the scheduling operation. Peng and Li (2020) adopted an adaptive multi-population method and defined a contribution degree according to the performance of the population, which can determine the probability of the population in obtaining the computational resources.

(b) Constructing composite solutions

In addition to using multi-populations, constructing composite solutions also can increase algorithm diversity. The composite solution is a special solution composed of several component solutions, which can also be regarded as a subpopulation. That is, the composite solution strategy actually implicitly divides the population into multiple subpopulations. Liu et al. (2010) introduced the composite solution strategy into PSO to propose a PSO with composite particles (PSO-CP) algorithm. In PSO-CP, each composite particle is designed to be a triangle shape and constructed by the remaining worst particle and two similar particles. To enable particles to continuously track the moving optima over time during the optimization process of DOP, the velocity-anisotropic reflection scheme that realizes information sharing within each composite particle is adopted in the PSO-CP. Liu et al. (2011) further enhanced the PSO-CP with a hyper-reflection mechanism. Herein, composite particles are constructed by a fitter particle and two randomly generated particles, which can bring more diversity.

(c) Designing novel solution update strategy

Besides, designing novel solution update strategy is another way to increase algorithm diversity by generating more new diverse solutions. It is efficient for almost all evolutionary computation algorithms, including GA, DE, and PSO. In GA, Tinos and Yang (2007) replaced the worst individual and individuals close to it with new randomly generated individuals in each generation. These new individuals provide diversity directly. Das et al. (2014) proposed an algorithm called dynamic DE with Brownian and quantum individuals (DDEBQ). There are three types of individuals in DDEBQ: the Brownian individual, the quantum individual, and other individuals. Different types of individuals adopt different methods to generate individuals. During the evolution, a new Brownian individual is generated within a Gaussian hyper-ellipsoid centered at the local best position, and a new quantum individual is generated within a specific region around the local best position, while other individuals evolve following a double DE mutation strategy. The Brownian and quantum individuals help in controlling the population diversity and thereby, enhancing the search efficiency. Zhu et al. (2018) proposed a replacement-based DE algorithm, which is based on the "DE/best/1" mutation operator. The novel replacement operator helps the population move towards the optima gradually. In PSO, the particle velocity and position update strategies are mainly considered when generating the new position of the particle. Du and Li (2008) proposed a new velocity updating strategy called differential mutation strategy in PSO to change the direction of a particle's velocity with certain probabilities and to extend the coverage of particle population to avoid being trapped into the local optimum. Cao et al. (2019) proposed a collaboration-based PSO algorithm by introducing a new particle guidance mechanism. That is, each particle is guided by a randomly selected particle and the best particle in the swarm to generate a new position. For each particle, instead of moving to this new position directly by itself, its newly generated position will be the position of the worst particle so long as the new particle has better fitness. This mechanism expands the search directions available to the target particle and enhances its exploration capability.

3.2.3 Accelerating convergence speed

When solving DOP, the algorithm needs to quickly locate the optimal solution in the new environment when the environment changes, which requires a fast convergence speed. Generally speaking, the consecutive dynamic environments often have strong relevance with each other. Therefore, the reuse of historical solutions has great potential to accelerate convergence speed in new environments. There are two ways to reuse historical solutions. One is reusing historical solutions directly and the other is predicting the location of the optimal solution and getting a promising initial population based on the historical solutions.

There may be some overlapping areas in the search space between two consecutive environments. Considering this, Cao et al. (2019) reused historical solutions directly. These historical solutions are composed of the best solution in each generation in the previous environment and form a trajectory of locating the optima in the search space. When the environment changes, some historical solutions will be put into the new initial population to estimate the promising optimal region. Liu et al. (2019d) proposed a dual-archive-based PSO, where the best solutions of each local region of past environments were stored in a fine-grained archive and a coarse-grained archive to preserve detailed information and systemic information, respectively. In the new environment, solutions stored in the fine-grained archive perform local search for purpose of exploitation, while solutions stored in the coarse-grained archive will be added into the new population for exploration. Wu et al. (2021b) proposed to directly put some found optimal solutions in the previous environment into the new initialized population in the new environment, so as to guide the population to locate better optimal regions faster.

Except for the direct reuse of the historical solutions, some algorithms predict promising solutions in the new environment based on the historical solutions. Woldesenbet and Yen (2009) proposed a variable relocation strategy (VRS) for dynamic evolutionary algorithm (RVDEA) to solve DOP. When the environment changes, VRS relocates individuals based on their changes in fitness values and the average sensitivities of their decision variables. Zhan et al. (2014) and Wang et al. (2016d) proposed two improved versions of RVDEA, called adaptive PSO (APSO) (Zhan et al. 2009) with VRS (APSO/VRS) and orthogonal learning PSO (OLPSO) (Zhan et al. 2011) with VRS (OLPSO/VRS), respectively. These two algorithms combine both advantages of APSO and OLPSO in optimization problems and VRS in relocating particles in DOP. Zhou et al. (2014) proposed a population prediction strategy combining the predicted center and estimated manifold to help initialize the whole population. Liu et al. (2018c) proposed a neural network-based change prediction (NNCP) method to discover the change law of the optima in different subareas and to predict new optima by training solution pairs. The training solution pair is composed of the local optima of each subarea from any two successive environments. Liu et al. (2020) also proposed another neural network-based method, called neural network-based information transfer (NNIT). In NNIT, the transfer model is trained by solutions collected from a new environment and its last environment and is used to generate new solutions in new environments to guide the population to approach the new global optimal region fast.

3.2.4 Extending application field

Many new EC-based algorithms have been proposed for specific application DOP by combining the advantages of different methods or by designing new strategies.

In solving path planning optimization problems in dynamic environments, Mahmoudzadeh et al. (2019) proposed a DE based multilayer framework to help unmanned underwater vehicle's long-duration missions in an uncertain dynamic environment. The multilayer framework includes a base layer of global path planning, an inner layer of local path planning, and an environmental sublayer, while DE is used by both base and inner layers. Liu et al. (2019a) introduced a novel learning transformation strategy to design an EDA-based learning fixed-height histogram algorithm to improve the accuracy and convergence speed. In solving dynamic flexible flow-shop scheduling problems, Tang et al. (2016) introduced a novel inertial weight inspired by the Hill functions into IW-PSO (PSO with interrelated weights) to minimize the makespan and energy consumption. In solving dynamic virtual machines (VMs) placement problems, Liu et al. (2017) introduced a dynamic pheromone deposition method and a special heuristic information strategy into the ant colony system (ACS)-based unified algorithm for the assignment and re-allocation of VMs. In the training neural networks field, Abdulkarim and Engelbrecht (2021) applied a dynamic PSO algorithm, specifically cooperative quantum PSO, to train neural networks in forecasting time series in dynamic environments. Experiments are conducted under four different dynamic scenarios and results show that dynamic PSO performs well in training neural networks.

3.3 EC for multimodal optimization problems

Multimodal optimization problems (MMOP) refer to optimization problems that have multiple optima. Generally speaking, research works to tackle the challenges for EC algorithms in solving MMOP can be categorized according to our function-oriented taxonomy into the following three aspects: *reducing problem difficulty* through multiobjectivization method, *increasing algorithm diversity* through niching method or novel operator, and *accelerating*

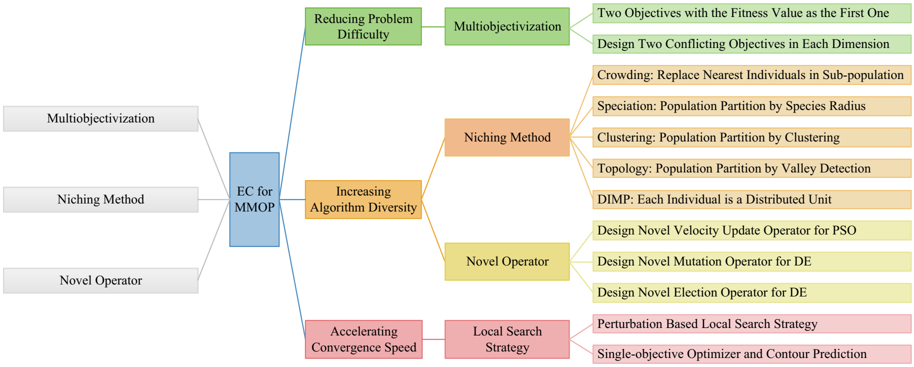
Multiobjectivization
Niching Method
Novel Operator
EC for MMOP
Reducing Problem Difficulty
Increasing Algorithm Diversity
Accelerating Convergence Speed
Multiobjectivization
Niching Method
Novel Operator
Local Search Strategy
Two Objectives with the Fitness Value as the First One
Design Two Conflicting Objectives in Each Dimension
Crowding: Replace Nearest Individuals in Sub-population
Speciation: Population Partition by Species Radius
Clustering: Population Partition by Clustering
Topology: Population Partition by Valley Detection
DIMP: Each Individual is a Distributed Unit
Design Novel Velocity Update Operator for PSO
Design Novel Mutation Operator for DE
Design Novel Election Operator for DE
Perturbation Based Local Search Strategy
Single-objective Optimizer and Contour Prediction

Fig. 6 Approaches of EC algorithms for solving MMOP

convergence speed through local search strategy. The detailed categories are shown as the right part of Fig. 6 and introduced as follows. Note that the left part of Fig. 6 is the taxonomy in some existing literature, like the other figures. In addition, a lot of research focuses on solving the real-world MMOP, which will also be introduced in the *extending application field* part in this section.

3.3.1 Reducing problem difficulty

There are multiple optima in MMOP and the fitness values of these optima are the same. In this case, the fitness value may not provide sufficient guidance to drive the evolution of the population, leading to the premature convergence of EC algorithms. To reduce the problem difficulty, some researchers attempt to transform MMOP into MOP by extracting information from the problem as another objective to provide more guidance to help the algorithm search easier. Therefore, this is regarded to make the problem simper by *multiobjectivization* (Wessing et al. 2013).

Usually, there are two optimization objectives in the transformed MOP where the fitness value of MMOP is directly adopted as the first optimization objective. For the second objective, some researchers proposed to adopt the gradient information (Yao et al. 2010; Deb and Saha 2012). This minimization objective is reasonable because the gradient of an optimum point is zero in the fitness landscape. However, we would like to note that these two optimization objectives do not conflict with each other. Instead, an optimum in MMOP can have the best fitness value and zero gradient concurrently. Therefore, such a multiobjectivization method may not provide a standard MOP. To deal with this problem, some researchers proposed to adopt distance metrics as the second optimization objective (Bandaru and Deb 2013; Basak et al. 2013). Specifically, they used the average distance of an individual to other individuals in the population and modeled it as a maximization objective. The distance metric can enhance the population diversity since it encourages the preservation of those individuals that are more different from others, which will help locate more optima. In recent research, Cheng et al. (2018) proposed adopting a grid-based diversity indicator as the second optimization objective. It contains two parts. The first part considers the distance between the individuals in the subpopulation. Herein, the individuals are normalized and mapped to a grid coordinate system and the subpopulation is defined according to an adaptive niche radius. The second part considers the number of individuals in a subpopulation. A larger distance and a smaller number of individuals are preferred as they indicate better population diversity.

Besides, different from the above multiobjectivization methods, Wang et al. (2015) designed two conflicting objectives in each dimension. As a result, it transforms an MMOP into *D* bi-objective optimization subproblems where *D* is the dimension size of the MMOP. A salient feature of such a transformation is that the optimal solutions in the MMOP are Pareto optimal solutions in its transformed MOP. Such a transformation method is ingenious but when dealing with those high-dimensional MMOP, i.e., *D* is large, there are a large number of bi-objective optimization subproblems, which are also very difficult for EC algorithms to solve.

3.3.2 Increasing algorithm diversity

A large quantity of research focused on enhancing the population diversity to help locate as many optima as possible in MMOP. These research works can be divided into two categories as *niching method* and *novel operator*.

(a) Niching method

Niching methods attempt to divide the population into a group of subpopulations, which can make the population evolve locally to maintain diversity. Crowding (Thomsen 2004) and speciation (Li et al. 2002; Li 2005) are two typical niching frameworks. The crowding framework compares the offspring with its nearest individual in the subpopulation and replaces the individual if it has better fitness. The size of the subpopulation is defined by a parameter named crowding size. The speciation framework divides the population where those individuals tracing the same optimum are expected to be in the same subpopulation. Then each subpopulation evolves independently to locate an optimum. In detail, a subpopulation consists of an individual and those individuals whose distances to it are within a predefined parameter named species radius. A challenging issue in crowding and speciation frameworks is that the algorithm's performance is highly sensitive to the parameter setting (e.g., the crowding size and the species radius) and it is very difficult to tune these problem-dependent parameters.

Besides the crowding and speciation for niching, clustering methods are also utilized to divide the population (Qu et al. 2012; Gao et al. 2014). They often require a parameter to guide the clustering (e.g., clustering number). Although this parameter is less sensitive than those in crowding and speciation, it may also influence the algorithm performance. Recently, Wang et al. (2020b) proposed an automatic niching DE (ANDE) algorithm that incorporated the affinity propagation clustering into EC algorithm to solve MMOP. In the ANDE algorithm, the niches are automatically formed by the affinity propagation clustering (Frey and Dueck 2007) that does not require the configuration of parameters like the number of clustering, and therefore the ANDE has a very powerful performance.

Moreover, some research utilized topological information to adaptively divide the population. In the fitness landscape, each optimum is a hill and there is usually a valley between two optima. Therefore, we can sample some new individuals between two individuals to detect whether there is a fitness valley and if so, the two individuals should be in different subpopulations (Stoean et al. 2010; Hui and Suganthan 2016). However, these sampling methods require extra function evaluations for valley detection. To avoid this problem, Li and Tang (2015) utilized the search history to take the place of sampling individuals. All the individuals in the history are stored in an archive and the algorithm selects some proper ones from the archive for valley detection.

Recently, Chen et al. (2020a) proposed a novel distributed individuals for multiple peaks (DIMP) framework to solve MMOP. The DIMP framework treats each individual as an independent unit to track an optimum and constructs a virtual population for this individual to assist its evolution, which can avoid the difficulties in dividing the population and also can maintain sufficient population diversity. In addition, the virtual population is adaptively controlled to help each individual sufficiently explore the search space and gradually converge to an optimum.

The comparisons of the features of different niching methods, mainly focusing on the niching strategy, niching parameter, and subpopulation overlap, are summarized in Table 3. It's noted that niching parameters here refer to direct niching parameters, such as niching radius, the subpopulation size, and the number of subpopulations. Subpopulations overlap means that the divided subpopulations have overlap individuals.

(b) Novel operator

Besides niching method, another popular method for enhancing the population diversity is to design novel operators for EC algorithms to adapt to MMOP. Usually, the researchers proposed to modify the operators in EC algorithms to help make the search more explorative to enhance population diversity. In conventional PSO, particles will learn from the same globally best solution, which will easily cause premature convergence when solving MMOP. Therefore, Qu et al. (2013) proposed a novel velocity update operator, in which the particle learned from neighboring particles to make the search locally around different peaks. In the domain of DE, a lot of research proposed novel mutation operators. Biswas et al. (2014) proposed a parent-centric normalized neighborhood mutation operator. An individual selects the other individuals for mutation according to a probability model constructed based on the distance between individuals. This new mutation operator helps to maintain the algorithm diversity by using well-defined local neighborhoods. They also proposed an improved information-sharing mechanism among individuals for inducing efficient niching behavior to increase the diversity of DE mutation operator (Biswas et al. 2015). Zhao et al. (2020) recently

Table 3 The comparisons of features of different niching methods

Reference	Niching strategy	Niching parameter	Subpopulations overlap
Thomsen (2004)	Crowding	✓	✓
Li (2005)	Speciation	✓	×
Stoean et al. (2010)	Sampling based topology	×	×
Qu et al. (2012)	Nearest Clustering	✓	✓
Gao et al. (2014)	Nearest Clustering	✓	×
Li and Tang (2015)	History based topology	×	×
Hui and Suganthan (2016)	Sampling based topology	✓	×
Wang et al. (2020b)	Affinity propagation clustering	×	×
Chen et al. (2020a, b)	DIMP framework	×	×

proposed a local binary pattern-based adaptive DE (LBPADE) that used the local binary pattern (LBP) information and a niching and global interaction (NGI) mutation operator. The LBP brings the idea from image processing that extracts relevant pattern information from the neighbors to identify the multiple regions of interest, so as to design an LBP-based niching method to locate multiple peaks in MMOP. Moreover, the NGI mutation operator incorporates information from both the niching and the global areas to increase algorithm diversity for effective exploration.

Although lots of works have been proposed to increase the diversity for generating solutions that can locate different optimal regions, like the above novel operators and niching method, it is strange that few works focus on the selection of individuals. If the selection operator is not well designed, diversity solutions in different optimal regions can not be selected into the next generation, even though these solutions have been generated. Therefore, a promising selection operator plays a significant role in solving MMOP. In this aspect, a representative work is proposed by Wang et al. (2018b) that introduced a novel selection operator for DE. It first conducts a clustering method on the combined set of the current population and newly generated individuals. Then the new population for the next generation is formed by selecting from each cluster by a probability model. By doing so, the new population is composed of individuals widely distributed in the search space, enhancing the population diversity.

3.3.3 Accelerating convergence speed

Besides locating multiple optima in MMOP, accelerating convergence speed to refine the accuracy of the found near-optimal solutions is also a challenging issue. A popular strategy is local search. Usually, the algorithms do a perturbation around a solution based on some probability models to improve its accuracy, in which the Gaussian probability model is the most widely used one. The general local search strategy on a solution using the Gaussian probability model can be formulated by using this solution as the mean and together with a standard deviation. Therefore, there are two challenging issues in the design of perturbation-based local search strategy. One is the setting of standard deviation and the other is the selecting of which solutions for local search.

For the setting of standard deviation, a fixed standard deviation in the Gaussian probability model is set (Yang et al. 2017c, 2017d). However, different optimization problems have different features and sizes of search space, so that a fixed value may not always do well. To deal with this problem, Wang et al. (2020b) proposed a setting scheme that gradually decreased the standard deviation during the evolutionary process. Chen et al. (2020a) proposed an exponential descent strategy. The standard deviation will adaptively decrease according to whether the current standard deviation continuously helps improve the solution accuracy.

For the selection of solutions for local search, some research selects solutions by probability models constructed based on the solution's fitness (Yang et al. 2017c, 2017d; Wang et al. 2020b). Solutions with better fitness have a higher chance to be selected. Chen et al. (2020a) adopted an archive to store the solutions with good fitness found during the evolutionary process. The clustering method is conducted on these solutions and the solution with the best fitness in each cluster is selected.

Besides the perturbation-based local search strategy, Cheng et al. (2018) used a single-objective PSO algorithm to perform local search inside a small region where an optimum may exist. Moreover, prediction is always a method to help accelerate the optimization speed. In a recent study, Wang et al. (2020b) adopted the idea of contour in geography into the EC algorithm to predict the potential optimal solutions, which was a promising way to accelerate the convergence speed.

3.3.4 Extending application field

MMOP is common in various fields in the real world. In the field of marketing, there are multiple Nash equilibria in electricity markets and Zaman et al. (2018) attempted to find them by EC algorithms. In the field of power system, Goharrizi et al. (2015) proposed a parallel MMOP algorithm to find multiple optima for the design of a voltage-source-converter-based high-voltage-direct-current transmission system. In electromagnetism, the design of permanent magnet synchronous machine also has multiple optimal solutions (Vidanalage et al. 2018). In mathematics, some research focused on finding multiple optima of nonlinear equation systems by using EC algorithms (Song et al. 2015; Gong et al. 2017). These studies widely show the potential of EC algorithms in solving the complex MMOP in real-world applications.

3.4 EC for multi-objective and many-objective optimization problems

Multi-objective and many-objective optimization problems are two kinds of complex continuous optimization problems, which typically involve two/three and more than three objectives, respectively. Compared with solving multi-objective optimization problems (MOP), solving many-objective optimization problems (MaOP) is more difficult because of the increase of the objective. When solving MOP and MaOP by traditional EC algorithms, it is so hard to maintain diversity and convergence on all objectives. On the one hand, as the number of objectives increases, the objective space becomes significantly large and the number of solutions that need to uniformly approximate the true Pareto front (PF) also increases rapidly. On the other hand, as the Pareto dominance relationship is defined based on many objectives, most of the solutions are non-dominated solutions while only a very small fraction of them are closed to the PF. Therefore, how to obtain well-distributed and approximately optimal solutions along the PF is a worth studying issue. Because of the similarity of solving MOP and MaOP, some approaches can be used to solve both MOP and MaOP. Thus, this part may cover both solving MOP and MaOP, but mainly focuses on solving MaOP by EC algorithms.

In order to solve MOP and MaOP more efficiently, many EC algorithms usually need to be enabled and enhanced with *increasing algorithm diversity* and *accelerating convergence speed*. Moreover, as the MOP and MaOP are difficult, *reducing problem difficulty* and *reducing running time* are also promising approaches to enable and enhance the EC algorithms in solving MOP and MaOP. The approaches of using EC algorithms to solve MOP and MaOP focusing on the above aspects are illustrated in the right part of Fig. 7. The left part of Fig. 7 is the traditional classification for multi-objective and many-objective evolutionary algorithms (Mane and Rao 2017). At last, approaches of EC algorithms in solving real-world MOP and MaOP are reviewed in the *extending application field* part.

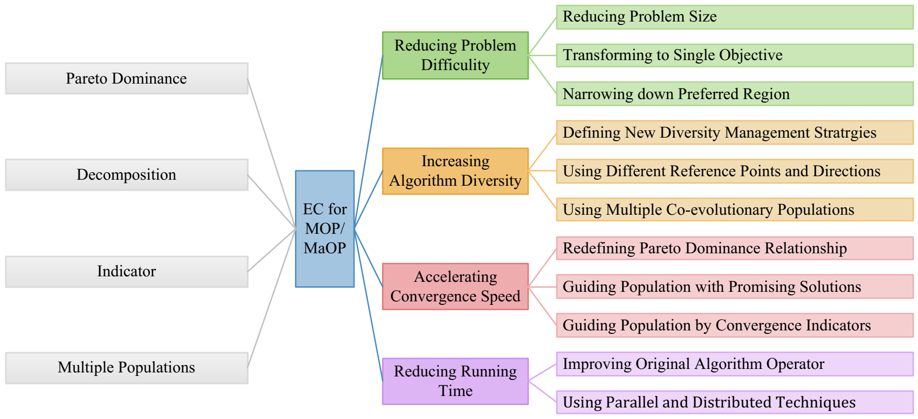
Pareto Dominance
Decomposition
Indicator
Multiple Populations
EC for MOP/ MaOP
Reducing Problem Difficulity
Increasing Algorithm Diversity
Accelerating Convergence Speed
Reducing Running Time
Reducing Problem Size
Transforming to Single Objective
Narrowing down Preferred Region
Defining New Diversity Management Stratrgies
Using Different Reference Points and Directions
Using Multiple Co-evolutionary Populations
Redefining Pareto Dominance Relationship
Guiding Population with Promising Solutions
Guiding Population by Convergence Indicators
Improving Original Algorithm Operator
Using Parallel and Distributed Techniques

Fig. 7 Approaches of EC algorithms for solving MOP/MaOP

3.4.1 Reducing problem difficulty

When solving a complex optimization problem, an obvious idea is to simplify the problem. When we apply this idea to solve MaOP, this idea is applied as reducing problem size, transforming to single objective, and narrowing down preferred region.

In MaOP, the number of objectives is typically more than three, but not all the objectives are conflicting. Therefore, reducing the redundant objectives can help reduce the size of MaOP, which is a straightforward way to reduce the difficulty of the problem. Accordingly, Singh et al. (2011) tried to omit the objectives one by one. If the number of non-dominated solutions obtained after omitting an objective does not change much, this objective is regarded as a redundant objective; otherwise, it is a relevant objective. This way, the complex MaOP can be reduced to a simpler MaOP with only the relevant objectives. Saxena et al. (2013) proposed a dimensionality reduction framework for linear and non-linear objectives using principal component analysis. However, these direct objective reduction methods may lead to losing the original dominance structure. Therefore, Cheung et al. (2016) proposed an objective extraction method to reduce the number of objectives and formulated the reduced objectives as a linear combination of original objectives. This method can retain the original dominant structure as much as possible to obtain the Pareto solutions of the original MaOP via solving the reduced MaOP. In other perspectives, Yuan et al. (2018) regarded the objectives reduction process as a multi-objective search problem and introduced three multi-objective formulations for objectives reduction. Very recently, Liu et al. (2021b) proposed to use diversity and convergence as two indicative objectives to reduce the MaOP to a bi-objective MOP, resulting in a new multi-objective framework for many-objective optimization (Mo4Ma).

Another intuitive method to reduce the problem difficulty is to transform MaOP into a/ some single objective optimization problem(s). Ishibuchi et al. (2009) used the scalarizing function with different weight vectors to aggregate the multiply objectives into a set of single objectives. This method decomposes the MaOP into different single objective subproblems and optimizes them simultaneously. However, the choice of scalarizing function has a significant impact on the search ability of the algorithm. Therefore, in the paper proposed by Ishibuchi et al. (2010), two scalarizing functions, the weighted Tchebycheff (Chebyshev) and the weighted sum, are applied simultaneously to enhance the performance of the algorithm. Besides, Li et al. (2017) decomposed a MOP into a number of single objective optimization problems and solved them in a collaborative way. It divides these sub-problems into several groups. At each generation, only one selected sub-problem from each group is optimized by CMA-ES, while other sub-problems are optimized by DE. Li et al. (2020a) transformed MaOP into a set of constraint single objective optimization problems where one objective was selected as the main objective to be optimized and the other objectives were transformed into constraints. Experimental results have shown the effectiveness of these methods for solving MaOP. However, these methods require a series of predefined well-distributed weight vectors. Differently, Hu et al. (2017) divided the process of solving MaOP into two stages. The first stage only finds several extreme solutions by using a decomposition-based single objective optimizer, which aggregates objectives into different single objective problems. Herein, the optimal solutions of the single objective optimization problems are namely the extreme solutions. The second stage uses a many-objective optimizer to approach the entire PF by extending the extreme solutions. Later, Sun et al. (2019c) proposed a new two-stage method with dynamic weight aggregation which could find convex, concave, linear, and mixed PF.

Furthermore, in order to reduce the problem difficulty, there is another worthy idea by narrowing down the preferred region. This method focuses on the regions preferred by decision-makers without exploring the entire PF since not all Pareto solutions are in line with the actual needs of decision-makers. It is only necessary to find the Pareto solution set for the regions that the decision-makers prefer. To this end, researchers bring the idea of focusing on the preferred region to reduce the problem difficulty to help solve both MOP (Thiele et al. 2009) and MaOP (Xiong et al. 2019). Thiele et al. (2009) projected the preference points onto the current Pareto set and found the neighbor-point set preferred by the decision-maker on the PF as the final solution set. Xiong et al. (2019) constructed the preferred regions with predefined information given by decision-makers. Reference points are generated in preferred regions to guide the solutions converge and distribute on the related part of PF.

3.4.2 Increasing algorithm diversity

As the MaOP is complex, many studies focus on increasing the diversity of the solutions to help search for the whole global optimal PF. The main approaches include: *defining new diversity management strategies*, *using different reference points and directions*, and *using multiple co-evolutionary populations*.

Defining new diversity management strategies into MaOP helps the EC algorithms produce well-distributed diverse solutions. Adra and Fleming (2011) proposed two density management (DM) mechanisms, named DM1 and DM2, to manage the diversity. Herein a spread indicator is defined to measure the spread of solutions. Then DM1 activates the diversity promotion strategy if the indicator is too small and DM2 adaptively adjusts the mutation range according to the spread indicator so as to increase the algorithm diversity. Recently, Zhang et al. (2020a) proposed an evolution strategy (ES)-based maximum extension distance strategy to increase the diversity of population. The proposed maximum extension distance strategy guides solutions to maintain the uniform distance and extension by simulating the distribution of isotropic magnetic particles in the magnetic field.

Also, as the search space expands, some methods tend to use different reference points and reference directions to increase the algorithm diversity. These methods define different reference points and reference directions which ensure the basic diversity of EC algorithms. Deb and Jain (2014) proposed the third version of non-dominated sorting GA (NSGA) algorithm, named NSGA-III, by predefining a set of well-distributed reference points and selecting solutions near the reference points during the selection operation to increase algorithm diversity. Similarly, Asafuddoula et al. (2015) proposed an improved decomposition-based EA that a set of uniformly distributed reference points was generated to ensure the diversity. Moreover, the method of using more different reference points and directions has been widely adopted by researchers to increase the diversity to promote MOP algorithms to solve MaOP like the dominance-based EA (Li et al. 2015b), reference vector guided EA (Cheng et al. 2016), strength Pareto EA (Jiang and Yang 2017), and decomposition-based EA (Cai et al. 2017). When facing more complex PF shapes in MaOP, He et al. (2019) proposed to increase the diversity for decomposition-based EA by dynamical decomposition strategy. The reference points are selected from the solutions themselves and adapted to the shape of PF automatically. Therefore, these different reference points can provide large diversity to guide the population to explore more different spaces.

Another promising way to increase algorithm diversity is using multiple co-evolutionary populations because different populations can be configured with different settings or they can cooperatively search for different regions. The multiple populations for multiple objectives (MPMO) were first proposed by Zhan et al. (2013) to provide a new framework for solving MOP. In Zhan et al. (2013)'s work, the MPMO framework was implemented in the PSO algorithm and a coevolutionary multi-swarm PSO (CMPSO) was proposed. The CMPSO assigns a population to each objective, which is a good way to avoid ignoring some objectives to ensure diversity. At the same time, an archive is adopted by CMPSO to integrate the information of multiple populations, which is conducive to share search information among multiple populations to increase diversity for better co-evolution. Later, the MPMO framework has become a new and efficient paradigm for solving MOP, which has been applied to other EC algorithms like DE (Wang et al. 2016b), ACS (Chen et al. 2019), artificial bee colony (Zhang et al. 2019a), and invasive weed optimization (Naidu and Ojha 2018), which have all obtained promising performance. Most recently, Liu et al. (2019c) proposed a multiple population-based coevolutionary PSO by extending the MPMO framework with a bottleneck learning strategy to solve MaOP, which could better ensure the diversity of the final solutions via MPMO and better ensure the convergence on all the objectives via a bottleneck objective learning strategy. Besides, Matos and Britto (2017) proposed a multi-swarm algorithm for MaOP. Specifically, different communication topologies are used in the swarms to co-evolutionary guide the particles, so as to maintain the diversity.

3.4.3 Accelerating Convergence Speed

Moreover, in order to make the solution approach the true PF as close as possible, studies focused on accelerating convergence speed are also significant in MaOP, mainly including *redefining Pareto dominance relationship*, *guiding population with promising solutions*, and *guiding population by convergence indicators*.

Due to the large number of objectives in MaOP, the number of non-dominated solutions obtained by the Pareto-based EC algorithm will enlarge exponentially because of the hard satisfaction of the Pareto dominance definition. However, only a small portion of the non-dominated solutions which are closer to the true PF can have a stronger ability to guide the population to convergence. This results in the dominance-resistance phenomenon (Purshouse and Fleming 2007) that leads to a decrease in selection pressure and affects the convergence performance of the Pareto-based EC algorithm. Therefore, it is necessary to improve or redefine the Pareto dominance relationship. Zou et al. (2008) proposed an L-optimal mechanism which was actually a subset of the Pareto set. Using the L-optimal mechanism can select more reasonable solutions for accelerating convergence speed. Yang et al. (2013) proposed a grid-based EA to convert Pareto dominance to grid dominance which could effectively reduce original non-dominated solutions to increase selection pressure. He et al. (2014) introduced a fuzzy mechanism to Pareto dominance by using a continuous function to quantify the degree of non-dominance between two solutions. Hence, solutions with a greater degree of non-dominance can be selected. Moreover, a new dominance relation (Yuan et al. 2016) and a strengthened dominance relation (Tian et al. 2019) were proposed to more strictly define only the best converged solutions as non-dominated, so as to push the population to converge faster.

There are also some studies that tend to select more promising solutions from the current Pareto set based on some specific preferred strategies to guide the population to converge faster. Wang et al. (2013b) pre-defined some points (called preference points). The preference points and the current solution will co-evolve. Solutions that dominate more preference points are regarded to be more promising. Li et al. (2014a) proposed a shift-based density estimation strategy to select more promising solutions that are more closed to the PF rather than scattered in the sparse region. Zhang et al. (2015b) considered the knee point in the current PF as a more promising solution. By paying more attention to the knee point, other solutions in the population can be guided to converge to the PF faster. More recently, Yu et al. (2021) used the α-dominance and knee-oriented dominance relationship to further identify the knee regions.

In addition, some studies propose to use performance metrics related to convergence as indicators to accelerate the convergence of the population. Sun et al. (2019b) used the inverted generational distance indicator while Shang and Ishibuchi (2020) used the hypervolume indicator to guide the convergence of algorithms. Moreover, Li et al. (2016a) used multiple indicators to get stronger convergence ability.

3.4.4 Reducing running time

The optimization and improvement of the original operator can effectively reduce the complexity and running time of the original algorithm. Srinivas and Deb (1994) proposed the original NSGA. The algorithm requires $O(MN^3)$ time complexity and $O(N)$ space complexity, where M denotes the objective numbers and N denotes the solution numbers. NSGA has improved by Deb et al. to the second version algorithm called NSGA-II (Deb et al. 2002). The new version reduces the computational complexity to $O(MN^2)$ by recording the dominant solutions and dominated solutions, which has also been widely used in MaOP algorithms. Wang and Yao (2014) used a corner sorting mechanism to rank the solutions, which could reduce the running time of determining the Pareto domination relationship. Zhang et al. (2015a) proposed an efficient non-dominated sort method that a solution only needed to be compared with those already in PF instead of all the solutions to determine whether it could be assigned to the PF. Based on this idea, they proposed a tree-based efficient non-dominated sort method that could reduce running time in solving MaOP (Zhang et al. 2016).

Moreover, some researchers propose to reduce execution time in indicator-based algorithms and decomposition-based algorithms. Considered the high execution complexity for hypervolume (HV) calculation, Bader and Zitzler (2011) used Monte Carlo simulation to approximate the exact hypervolume values and got the rankings of solutions induced by the HV indicator, and then got a balance between the accuracy of the estimates and the available computing resources. Jiang et al. (2015) measured the HV contribution of a solution by considering partial solutions rather than the whole solution set, and thus proposed a simple and fast HV indicator-based multi-objective evolutionary algorithm that can reduce the high time complexity in measuring the exact HV contributions of different solutions. Deng and Zhang (2019) transformed the hypervolume into an (m–1)-D (where m is the number of objectives) integral by using polar coordinate, and then proposed two approximation methods for computing the HV and HV contributions.

Another way for reducing running time is paralleling and distributed computing (Luna and Alba 2015; Talbi 2019). To reduce the running time for fitness evaluation, the evaluations of the whole population are concurrently assigned by several processors in every iteration (Depolli et al. 2013). Li et al. (2015a) proposed a parallel version of multiobjective PSO-based decomposition algorithm by using both message passing interface and OpenMP. This algorithm combines distributed-memory and shared-memory programming models and can fully use the processing power. Experimental results show a speedup of 2 × by using this algorithm. Campos et al. (2019) proposed a parallel strategy for PSO to deal with multiple populations to solve both MOP and MaOP. Specifically, the multiple populations evolve on the independent processors parallelly, so as to reduce running time. Besides, the communication among populations is established by a fully-connected network to share search information.

3.4.5 Extending application field

Many practical applications have spawned the development of MaOP. Researchers in different fields use EC algorithms as basic tools to optimize practical MaOP. For example, in the industrial field, Cheng et al. (2017) viewed the hybrid electric vehicle control model as a MaOP with different objectives, such as fuel consumption, battery stress, emissions, and noise. Bitsi et al. (2019) used a Pareto-based many-objective optimization algorithm to optimize conflicting objectives such as torque capacity, efficiency, and torque density of automotive motors to design a better motor topology. Salimi and Lowther (2016) used a projection-based objective reduction method to optimize the five-objective industrial motor design problem. Furthermore, in order to improve the design robustness of axial-flux permanent magnet synchronous generators under uncertain factors, Sabioni et al. (2018) considered efficiency, material cost, weight, diameter, and efficiency maximum estimated deviation as the optimization objectives. Martins et al. (2019) used NSGA-II to optimize the parameters in the voltage-controlled oscillator to help it adapt to different consumption requirements of the Internet of things and cellular applications.

In some other subject areas, Starkey et al. (2019) added fuzzy logic systems to the algorithm to get tradeoff solutions of workforce resource allocation, aiming at optimizing the workforce resource allocation of large organizations or companies. Fleck et al. (2017) extended the NSGA-III algorithm to solve model transformation modularization problems. Miranda et al. (2018) focused on the performance of classifiers after training in imbalanced data set and modeled the selection of sampling strategies in imbalanced data set as a MaOP, due to that different sampling strategies lead to different training performances of the classifier. In order to obtain an ensemble classifier with balanced performance, Asafuddoula et al. (2018) used the accuracy of different classifiers as optimization objectives.

Moreover, as some kinds of new paradigms for solving MOP and MaOP have shown efficient ability in solving benchmark problems, they have been widely extended to real-world practical problems. An example is the popularization of the MPMO framework (Zhan et al. 2013). Li et al. (2014b) used the MPMO framework for solving multi-objective power system economic dispatch. Yao et al. (2017) used the MPMO framework for solving multi-objective workflow scheduling in a cloud system. Chen et al. (2019) proposed a multi-objective ACS based on the MPMO framework for optimizing both the execution time and executing cost in cloud workflow scheduling problems. Zhou et al. (2020) modeled the airline crew rostering problem as a bi-objective optimization problem that aimed at optimizing both the fairness and satisfaction of crew, and they extended the MPMO framework to efficiently solve the proposed model. Zhao et al. (2021) proposed to use the MPMO framework for solving multi-objective cardinality constrained portfolio optimization problems. Liu et al. (2021a) proposed to use the MPMO framework-based multi-objective PSO to solve the emergency resource dispatch problem.

3.5 EC for constrained optimization problems

The constrained optimization problems (COP) are a kind of complex problem that should satisfy a set of constraints. Therefore, there are lots of infeasible solutions in the search space because they do not satisfy one, some, or all of the constraints. Only a small fraction of the search space contains feasible solutions. Due to the powerful search ability of EC algorithms, many researchers have focused on using them to solve COP in the recent decades. However, it is difficult to determine which solution is better because the COP has infeasible regions where the solutions may have good fitness values. We need to combine constraint-handling techniques (CHTs) when using EC to deal with COP. The EC algorithms are served as the search engine, while CHTs guide the algorithms on how to select the solutions for the population of the next generation. The classic taxonomy of technologies for solving COP is shown in the left part of Fig. 8. However, in this paper, we classify the literature of EC algorithms in solving COP according to our function-oriented taxonomy from three aspects, which are *reducing problem difficulty*, *increasing algorithm diversity*, and *accelerating convergence speed*. Finally, the *extending application field* approach for COP will also be given. The illustration of using EC to solve COP can be seen in Fig. 8.

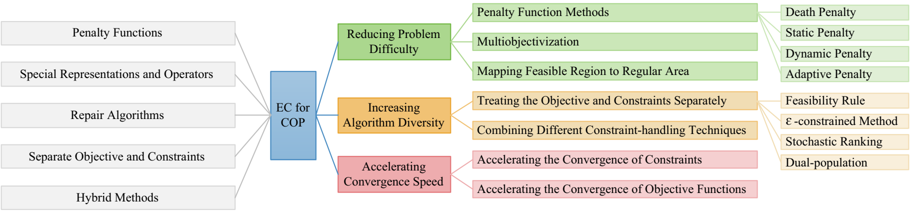
Penalty Functions
Special Representations and Operators
Repair Algorithms
Separate Objective and Constraints
Hybrid Methods
EC for COP
Reducing Problem Difficulty
Increasing Algorithm Diversity
Accelerating Convergence Speed
Penalty Function Methods
Multiobjectivization
Mapping Feasible Region to Regular Area
Treating the Objective and Constraints Separately
Combining Different Constraint-handling Techniques
Accelerating the Convergence of Constraints
Accelerating the Convergence of Objective Functions
Death Penalty
Static Penalty
Dynamic Penalty
Adaptive Penalty
Feasibility Rule
ε -constrained Method
Stochastic Ranking
Dual-population

Fig. 8 Approaches of EC algorithms for solving COP

3.5.1 Reducing problem difficulty

Considered the difficulty of finding feasible regions in the search space, a good idea is to reduce the problem difficulty of COP. Methods for reducing the problem difficulty of COP can be divided into three categories, namely *penalty functions methods*, *multiobjectivization*, and *mapping the feasible region into a regular area*.

The penalty functions method is a well-known method for reducing problem difficulty because it is very simple and intuitive to be used to transform COP into an unconstrained problem. It adds the constraint violation as the penalty factor to the fitness function, so that the objective and constraint functions can be considered in a single function simultaneously. However, the penalty factor is problem-dependent, which is hard to be set with an appropriate value. According to the different adjustment methods of the penalty factor, penalty functions methods can be divided into the following four categories. Firstly, the simplest penalty function method is to directly set the penalty factor as positive infinity, which is called "death penalty" (Yeniay 2005). It rejects all infeasible solutions into the population, and then losing information of infeasible solutions. Thus, this method may spend much time to find a feasible solution. Secondly, static penalty functions use the fixed penalty factor during the evolutionary process. Hsieh et al. (2015) proposed a two-phase immune evolutionary algorithm to solve nonlinear COP, based on the static penalty function method. However, the importance of the penalty factor is often different during evolution, so it is inappropriate to keep the fixed value. Thirdly, dynamic penalty functions change the penalty factor during the evolutionary process (Liu et al. 2018a). Liu et al. (2016a) proposed an S-type function to set the penalty factor, which was only adjusted by the current number of iterations. Since the penalty factor is generally problem-dependent, the same adjustment of the penalty factor may be unreasonable for all COP. Fourthly, adaptive penalty functions adjust the penalty factor according to the feedback information of the population. Hamida and Schoenauer (2002) proposed to adjust the penalty factor based on the comparison between the expected feasible ratio and the feasible ratio of the current generation. If the current feasible ratio is less than the expected feasible ratio, the penalty factor increases. Tessema and Yen (2009) proposed a self-adaptive penalty (SP) function that combined modified fitness value and the penalty. The penalty is the weighted sum of the modified fitness value and constraint violation and the weight is determined by the current feasible ratio. Saha et al. (2016) proposed a fuzzy rule-based penalty function, where the input of the Mamdani-type fuzzy inference system was associated with constraint violation, the objective function value, and the current feasible ratio.

Another popular method for reducing the difficulty of COP is *multiobjectivization*, which transforms a COP into a MOP. It treats the overall constraint violation as another objective or each constraint function as an objective so that the algorithm can easily find feasible solutions for the MOP (Wang and Cai 2012a, b).

In addition, there are also some studies for reducing the difficulty of COP by mapping the feasible region into a regular area that is easy to handle. For example, Koziel and Michalewicz (1999) proposed a homomorphous mapping method that transformed the feasible region into a regular *n*-dimensional cube, which was easier to facilitated EC algorithms to optimize. However, this kind of mapping process itself is complex and requires additional computational cost.

3.5.2 Increasing algorithm diversity

In COP, some infeasible solutions may be closer to the feasible optimal solution than some feasible solutions. Therefore, how to select solutions for the next population is the key issue of COP, and it is of great significance for COP to improve the diversity of algorithms. There are mainly two methods to increase algorithm diversity in current studies. One is *treating the objective and constraints separately*, and the other is *combining different CHTs with different characteristics*.

It is intuitional to treat the objective and constraints separately and the relevant methods can be divided into the following four categories. The first category is the common method based on the feasibility rule (FR). Generally speaking, the FR is with three criteria proposed by Deb (2000): (1) Any feasible solution is better than any infeasible solution; (2) In two feasible solutions, the one with a better objective function value is preferred; and (3) In two infeasible solutions, the one with smaller constraint violation is preferred. Under the FR method, Zhang et al. (2014) solved the COP by proposing to use an artificial immune system for optimizing the objective and the FR as the CHT for dealing with the constraints, respectively. The FR is simple and easy to implement, but the feasibility is overemphasized, making it easy to fall into a local optimum. Therefore, some works enhance the FR comparison by additionally adopting objective function value comparison for further increasing the algorithm diversity. For example, Wang et al. (2016c) proposed to incorporate the fitness value comparison into the FR, where the losing offspring based on FR comparison with a better fitness value could be stored into an archive. In the replacement mechanism, the solution with the maximum constraint violation in each group of the population and the solution with the minimum constraint violation in the archive are compared based on the fitness value.

Besides the FR, the second category for the separation of the objective and constraints is the ε-constrained method proposed by Takahama and Sakai (2006). Since some infeasible solutions may carry important information, they introduced a parameter ε to relax the constraints comparison in the FR (Deb 2000). A solution with constraint violations not greater than ε can be regarded as a feasible solution. Under this consideration, the ε–based FR can be accommodated to select a solution with better objective function value and ε constraints satisfaction. The setting of ε directly controls the balance between the objective function and constraint violation. In the original ε-constrained method, as the number of iterations increases, ε decreases as a power function, where the power is fixed (Takahama and Sakai 2006). Later, the same authors proposed to control the ε decreasing by a power function, where the power was dynamically changed (Takahama and Sakai 2010). Furthermore, Fan et al. (2019) proposed to use the current state of the population to guide the adaptive change of ε. In the early stage of evolution, if the current feasible ratio is small, ε decreases exponentially to help explore in the feasible region. When the feasible ratio is large enough, ε will relax to a large value to enhance the exploration in the infeasible region.

The third category for separating objectives and constraints is stochastic ranking (SR) proposed by Runarsson and Yao (2000), which compares two solutions only based on the objective function or constraint violation. It introduces a fixed parameter to determine the probability of comparison using the objective function.

The fourth category to separate constraints from the objective is dual populations proposed by Gao et al. (2015). When the number of feasible and infeasible solutions is greater than three, the population is divided into two subpopulations that only optimize constraints and the objective function, respectively. When implemented in DE, the two subpopulations cooperate and share information by the guidance solution in the mutation operator.

Besides the objective and constraints separation method, another popular method for increasing algorithm diversity is to deal with constraints by combining CHTs with different characteristics. According to the no free lunch theorems (Wolpert and Macready 1997), no single CHT can effectively deal with all COP. Different CHTs favor different types of solutions, so as to increase the diversity of solutions. Wang et al. (2008) designed an adaptive tradeoff model for three situations. In the infeasible situation, a multi-objective optimization method is adopted to handle constraints, while in the semi-feasible situation, an adaptive penalty function method is used. As for the feasible situation, the COP is reduced to an unconstrained optimization problem and only the objective function value is considered. Mallipeddi and Suganthan (2010) proposed a framework for COP based on an ensemble of four CHTs, i.e., SP (Tessema and Yen 2009), FR (Deb 2000), ε-constrained method (Takahama and Sakai 2006), and SR (Runarsson and Yao 2000). It uses multiple populations with multiple CHTs and all populations communicate by sharing their offspring. Hellwig and Beyer (2018) proposed a matric adaptation evolution strategy by combining successful constraint handling techniques, the ϵ-level ordering, and gradient-based repair, with Matric Adaptation Evolution Strategy. Both approaches are successful constraint handling techniques, which can help the population move towards feasible regions. Wang et al. (2019a) proposed the composite DE with three different trial vector generation strategies. One of three strategies is the random generation strategy, which contributes to diversity. This approach also combines with the FR and ε-constrained method to select solutions for the next generation population.

3.5.3 Accelerating convergence speed

When solving COP, how to quickly converge to the feasible region is a problem worth thinking about. The CHTs discussed above are generally used for comparison between two solutions. However, in recent years, some researchers have turned to use CHTs to directly guide the solution generation. This kind of method mainly focuses on the mutation operation to accelerate convergence speed for solving COP. When talking about accelerating the convergence speed of solving COP, there are two methods in current studies. One is *accelerating the convergence of constraints*, and the other is *accelerating the convergence of the objective functions*.

Since the feasible region of some COP with equality constraints may be small, it is difficult to obtain a feasible solution in this case. Finding a feasible solution quickly is one of the important ways to accelerate the convergence speed of algorithms. The gradient-based mutation (Takahama and Sakai 2006) tries to repair the infeasible solutions by the gradient of constraints. Hamza et al. (2016) proposed the mutation based on constraint consensus, whose calculation was also related to the gradient of unsatisfied constraints. The above two gradient-based methods can accelerate the search for feasible solutions. Maesani et al. (2016) proposed the memetic viability evolution, in which viability boundaries on each constraint were defined separately in the local search units. Thus, additional information on each constraint can be gathered to help search faster. Spettel et al. (2019) devised a special mutation operator that did not violate any linear constraints and repaired the non-negativity constraints by projection.

However, the above methods only consider speeding up the process of finding a feasible solution, but without the attention to the convergence of the objective function. It is also significant to accelerate the convergence speed within the feasible region. Gong et al. (2015a) proposed an adaptive ranking mutation operator based on DE, which sorted the solutions according to different criteria in different situations. It utilizes better solutions to guide the search, so that the ability of exploitation can be improved to gain faster convergence speed. This algorithm contains two different trial vector generation strategies for accelerating convergence speed, which is different from the random generation strategy used for increasing algorithm diversity in composite DE proposed by Wang et al. (2019a). One of the trial vector generation strategies is guided by constraint violation and the other is guided by the objective function. In another work, Wang et al. (2020a) proposed to explore the correlation between constraints and the objective function in the early learning stage and made use of the correlation in the evolving stage to help the algorithm converge faster.

3.5.4 Extending application field

As many COP come from real-world applications, there are also many EC algorithms proposed to solve real-world COP. In the automatic control field, Wang et al. (2020a) applied the DE-based algorithm to the gait optimization of humanoid robots, which was restricted by the rotation angle of each degree of freedom. Xu et al. (2019) proposed a new CHT that divided the population into different sections and applied it to the constrained parametric optimization for a breast cancer immunotherapy model. As for resource scheduling, Chen and Chou (2017) proposed the NSGA-II to solve airline crew roster recovery problems, which had a set of constraints for safety. Zheng and Wang (2018) proposed a collaborative fruit fly optimization for the resource-constrained unrelated parallel machine green scheduling problem. In cloud computing, Chen et al. proposed the dynamic objective GA (Chen et al. 2015a) and ACS (Chen et al. 2015b) to solve the deadline constrained cloud computing resources scheduling. Liu et al. (2018b) proposed an ACS-based method for virtual machine placement, which was constrained by the resource requirement. In electromagnetism, Kovaleva et al. (2017) proposed a cross-entropy method for electromagnetic optimization with constraints.

3.6 EC for expensive optimization problems

Expensive optimization problems (EOP) refer to optimization problems that require computational expensive simulation or calculation to evaluate candidate solutions (Jin 2005; Shan and Wang 2010; Tenne and Goh 2010). Due to the expensive evaluations, these problems are difficult for EC algorithms to optimize (Jin 2011). So far, more and more EC-based research has been proposed for dealing with EOP. Generally speaking, to make EC

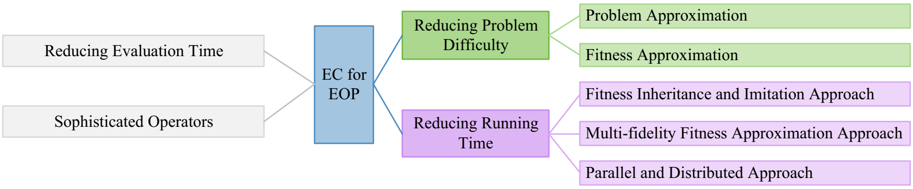
Reducing Evaluation Time
Sophisticated Operators
EC for EOP
Reducing Problem Difficulty
Reducing Running Time
Problem Approximation
Fitness Approximation
Fitness Inheritance and Imitation Approach
Multi-fidelity Fitness Approximation Approach
Parallel and Distributed Approach

Fig. 9 Approaches of EC algorithms for solving EOP

algorithms work efficiently on EOP, there are two major general approaches to make the algorithms obtain acceptable solutions within acceptable time, i.e., *reducing problem difficulty* and *reducing running time* of performing fitness evaluations. In addition, the *extending application field* approach can also provide valuable insight for solving expensive application optimization problems. Therefore, this part surveys the related works for solving EOP classified according to our function-oriented taxonomy, as shown in Fig. 9.

3.6.1 Reducing problem difficulty

In many optimization problems, mathematical or exact fitness functions for evaluating candidate solutions may not exist (Jin 2005). In such situations, the solutions can only be evaluated by computationally expensive numerical simulations or physical experiments, e.g., wind tunnel experiments. This poses great challenges on EC algorithms because most EC algorithms are based on the fitness evaluations for evolution. To solve this problem and reduce the optimization difficulties, approximation methods have been widely researched (Shan and Wang 2010; Tenne and Goh 2010; Jin 2011; Jin et al. 2019). Generally speaking, existing approximation methods for reducing the problem difficulty can be mainly classified into two categories as *problem approximation* and *fitness approximation* (Jin 2005), which are described as follows.

Problem approximation is a straightforward way to employ an approximated problem to replace the original problem. The new problem is approximately the same as the original problem but is much easier to solve. Nguyen et al. (2017) proposed two strategies to design simplified simulation models to replace the original simulation for the design problem of dispatching rules. The first strategy is to reduce the warmup and running time of simulations, while the second strategy is to reduce the search space, e.g., reduce the number of machines and the number of operations per job. The experimental results show that combining these two strategies is able to generate a simplified but accurate enough simulation model that requires less computational cost than the original simulation evaluations. Voutchkov et al. (2005) proposed a simplified mathematical model, which greatly reduced the expensive computational cost for solving sequential combinatorial finite element problems. The mathematical model is developed based on prior knowledge and assumptions and is optimized by a few real finite element simulations. This model has shown to be a fast, effective, and general method for replacing computationally expensive finite element models. Furthermore, in some real-world problems, there have been different kinds of fitness evaluations that can be adopted to simplify the problem. For example, in the optimization of aerodynamic structures, 2-*D* and 3-*D* computational fluid dynamics (CFD) simulation can be used to replace the real-world wind tunnel experiments, where the latter is much more computational expensive (Jin and Sendhoff 2009).

Different from the problem approximation methods that approximate the original evaluation procedure, the fitness approximation is to directly approximate the fitness value of candidate solutions. Given some evaluated data, i.e., solutions and their corresponding fitness values, EC algorithms can adopt fitness approximation methods to approximate the objective function. After this, the approximated objective functions, which are also called surrogates in the literature, can be employed to replace the original objective functions and evaluate new candidate solutions to drive the EC algorithms (Jin et al. 2019). Generally speaking, the surrogates, i.e., the approximated objective functions, are much easier to be calculated than the original objective functions and therefore they indeed reduce the difficulty of EOP. As the fitness approximation only needs to focus on the mapping between solutions and their fitness, it does not need to rely on the prior knowledge and specific characteristics of the problem, so many methods have been researched and proposed to build surrogates, including traditional interpolation methods (Zhou et al. 2005) and machine learning techniques such as Kriging model (Buche et al. 2005; Chugh et al. 2018), artificial neural networks (Jin et al. 2002; Willmes et al. 2003; Jin and Sendhoff 2004), radial basis function neural networks (Lim et al. 2010; Martinez and Coello 2013; Regis 2014; Sun et al. 2017), and random forest (Sun et al. 2020; Wang and Jin 2020). Furthermore, effective learning strategies have also been studied to improve the accuracy of fitness approximations. Wang et al. (2017) proposed a committee-based active learning method for building accurate surrogates. Wang et al. (2019b) employed an ensemble learning method to generate and combine a set of surrogates to improve the accuracy of functional predictions. In addition, Wei et al. (2021) proposed a level-based learning strategy and the gradient boosting gradient classifier to improve the robustness and scalability of surrogates for approximating high dimensional expensive problems. In the data-driven EA (DDEA), the number of true evaluated solutions is crucial for building good surrogates. However, if too much data is used, the problem becomes more complex. Therefore, how to build promising surrogates within few-shot true evaluated data is an important research topic. To this aim, very recently, Li et al. (2020c) proposed to build an accurate surrogate by efficiently manipulating the small number of available data via a localized data generation (LDG) method, so as to design a boosting DDEA with LDG (BDDEA-LDG). Moreover, Li et al. (2020d) further proposed a DDEA with perturbation-based ensemble surrogates (DDEA-PES) on the small number of available data. The few-shot true evaluated data makes the algorithms slighter and easier for using.

3.6.2 Reducing running time

As the computational cost is large in EOP, it is necessary to reduce the running time to make the EC algorithm applicable in real-world EOP. Generally speaking, there are three approaches to reduce the running time of expensive fitness evaluation. The first one is the *fitness inheritance and imitation* approach, the second one is the *multi-fidelity fitness approximation* approach, and the third one is *parallel and distributed* approach.

Firstly, as the fitness evaluation process is expensive, the fitness inheritance and imitation approach approximate the individual fitness based on other individuals instead of executing the fitness function. In fitness inheritance, the fitness of a new individual inherits from its parents. Chen et al. (2002) proposed a fitness inheritance method to speed up the algorithms. In addition, Sastry et al. (2001) provided analyses of the fitness inheritance in EAs. Different from fitness inheritance, fitness imitation estimates the fitness value of an individual based on some other evaluated individuals. Salami and Hendtlass (2003) proposed a fitness imitation method where the fitness of a new individual was the weighted sum of the fitness of the evaluated individuals. Tian et al. (2016) proposed a Gaussian similarity measurement to better estimate individual fitness. Kim and Cho (2001) divided the population into several clusters and calculated most of the individual fitness through estimation. Similarly, Jin and Sendhoff (2004) proposed a cluster-based method to reduce the fitness evaluations, where the individual closest to the cluster center was evaluated by real fitness evaluation while the rest were evaluated by estimation. Besides, Sun et al. (2013) proposed a fitness estimation strategy to estimate the individual fitness based on the distance between this individual and the evaluated individuals. Recently, the fitness estimation is further adopted for choosing individuals for real evaluations appropriately, which enhances the optimization accuracy and efficiency (Sun et al. (2017)).

Secondly, besides the fitness inheritance and imitation approach, the multi-fidelity fitness approximation approach has also been researched to reduce expensive evaluations. In many real-world applications, the fidelity of fitness evaluation can be manually controlled and there is a trade-off between fitness fidelity and computational cost (Wang et al. 2018a). In such a situation, multi-fidelity fitness approximation considers how to better use simulations and models with multiple fidelities to search for the optimal solutions, so as to obtain acceptable solutions within a much shorter time. Wang et al. (2018a) proposed three strategies in PSO to adaptively select an appropriate low fidelity model to evaluate individuals, which could reduce the running time. Also, Li et al. (2016c) adopted high-fidelity surrogates for problem decompositions and then employed computationally cheap surrogates to replace the computational expensive evaluations to drive the EC algorithms. In addition, Sun et al. (2017) combined two different approximation methods, i.e., the surrogate-assisted approximation and individual-based approximation, to reduce the expensive evaluations in high dimensional expensive problems. Wu et al. (2021c) proposed a novel scale-adaptive fitness evaluation (SAFE) method to adopt some evaluation methods with different time cost and corresponding accuracy scales to complete the fitness evaluation process cooperatively and efficiently and obtained promising results. The SAFE mainly adopts fitness evaluation methods with small time cost (but with lower accuracy) in the early stage to save computational time and can adaptively switch to other fitness evaluation methods with higher accuracy in the later stage to increase solution accuracy. This way, the algorithm can make the best balance between computational time and solution accuracy, resulting in promising results by reducing running time. Zhang et al. (2021) proposed multifidelity-based surrogate models to replace the exact expensive fitness evaluation, which have shown better performance than the compared algorithms within the same training time.

Thirdly, when evaluating candidate solutions is computationally expensive, parallel and distributed techniques can be employed to reduce the computational time cost. Therefore, many methods have been proposed to speed up EC algorithms through distributed techniques for solving EOP (Gong et al. 2015b). Liu et al. (2016b) proposed a parallel DE for solving the power electronic circuit (PEC) optimization problem, which required time-consuming simulations for fitness evaluations. Based on the distributed cloud computing resources, the parallel DE can assign candidate solutions to different resources for parallel evaluations, so that the time cost can be reduced. Considering that the computational resources may have different workloads and computing performance, Ma et al. (2017) further proposed a distributed DE with a load balance strategy to allocate the computational tasks to different resources efficiently and appropriately. In this method, the load information of each resource is calculated and considered for performing dynamic resource allocations, so that the topology (i.e., mapping) between the individuals and the resources can change adaptively for higher utilization of the concurrent computational resources. Very recently, Liu et al. (2021c) further studied the resource-aware technique to make the computational resources be allocated with suitable tasks, so as to propose a much faster distributed DE for training expensive neural-network-based controller in PEC. Different from only considering the load information of all resources, the Cloudde proposed by Zhan et al. (2017) calculates the appropriate number of machines or computing nodes for handling each task and then allocates resources accordingly, which is shown to be very promising and efficient in reducing the running time of evaluating PEC.

In addition to employing parallel and distributed techniques to accelerate the real fitness evaluations, parallel and distributed techniques have also been integrated with surrogate models. Karakasis et al. (2003) proposed a distributed GA with surrogate models to reduce the computational time cost of calling excessive computational fluid dynamics in an aerodynamic shape optimization problem. The algorithm partitions the population into several subpopulations and each subpopulation is evolved with surrogates concurrently. The best solution in each subpopulation will be regularly migrated to other subpopulations. The experimental results show that the combination of surrogates and distributed methods results in maximum computational efficiency in terms of the time cost. Akinsolu et al. (2019) proposed a parallel model assisted EA (PSMEA) to better parallelize the surrogate-assisted EAs. In PSMEA, several diverse solutions with promising approximated fitness are selected at the same time for the real fitness evaluation. During this process, the evaluation of each selected solution can be performed concurrently, which can reduce the expensive computational time. Furthermore, if there are multi-fidelity or multi-level surrogates, the algorithm parallelization can be further enhanced. Karakasis et al. (2007) proposed a hierarchical distributed EA for shape optimizations, where each subpopulation would concurrently employ a surrogate with different accuracies and time costs. The better solutions will be moved to the subpopulation using the surrogate with higher accuracy and higher time cost while the worse solutions will be moved to the subpopulation using the surrogate with lower accuracy but cheaper time cost. In addition, Sun et al. (2019d) proposed an asynchronous parallel surrogate optimization algorithm, which was based on an ensemble surrogating model and stochastic response surface method. In this algorithm, parallel computing technology is adopted to accelerate both the weights update of ensemble surrogates and the parameter optimizations, which has shown to be efficient.

3.6.3 Extending application field

As many EOP are almost from real-world applications, a lot of research has been conducted to deal with these problems by considering the problem types and characteristics, resulting in application-driven EC algorithms. In the health care field, Wang et al. (2016a) proposed a novel EA to optimize the design of trauma systems, where the fitness evaluation involving a large number of incidents was very computationally expensive. The authors found that the problem is an off-line data-driven optimization problem and proposed a novel EA with a clustering method and multi-fidelity surrogates to make a trade-off between computational time and optimization accuracy. The experimental results show that the proposed algorithm can save about 90% of computational budgets to produce an acceptable solution. In the fused magnesium furnaces optimization field, Guo et al. (2016) considered that the fitness evaluation was expensive to perform but only a small amount of evaluated data could be used to build the surrogate to approximate the fitness evaluation in such a typical small data-driven optimization problem. They constructed a low order polynomial model to generate more data to help approximate the fitness evaluation better. In the transonic airfoil design optimization field, Wang et al. (2017) proposed a surrogate-assisted EC algorithm with active learning. Although the evaluation of airfoil performance requires time-consuming CFD simulations, the proposed algorithm can obtain a high-quality solution efficiently through active learning-based surrogates. Similarly, Li et al. (2020d) applied GA with perturbation-based ensemble surrogates for the expensive airfoil design optimization as well and obtained great results. In an expensive shape optimization of an air intake ventilation system, Chugh et al. (2017) proposed a Kriging-based EA to tackle three

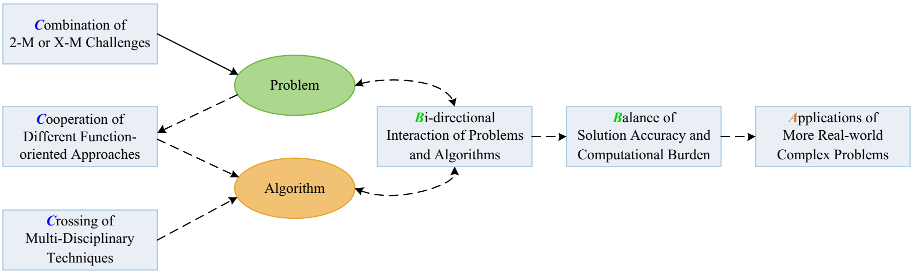
Combination of 2-M or X-M Challenges
Cooperation of Different Function-oriented Approaches
Crossing of Multi-Disciplinary Techniques
Problem
Algorithm
Bi-directional Interaction of Problems and Algorithms
Balance of Solution Accuracy and Computational Burden
Applications of More Real-world Complex Problems

Fig. 10 The future 3C-2B-1A research directions and their relationship

main challenges, i.e., formulating the optimization problem, connecting different simulation tools, and dealing with computationally expensive objective functions, which obtained promising results in the blast furnace optimization problem. Guo et al. (2020) constructed two surrogate models with support vector regression so as to efficiently evaluate supporting quality for the bolt supporting networks optimization. Besides, Li et al. (2020c) applied GA with boosting surrogate ensembles for the arterial traffic signal timing real-world expensive optimization problem, which can efficiently reduce traffic congestions.

4 Potential research directions and open problems

Based on the above survey, it is sure that using EC algorithms to solve complex continuous optimization problems has become a new and rising trend in both the EC community and the complex system community. In this section, we discuss some potential research directions and open problems, hoping to inspire wider and deeper research works in this field.

The discussions are based on two main bodies. That is, the “Problem” and “Algorithm”, as shown in Fig. 10. From the problem aspect, future research surely includes the “*Combination of 2-M or X-M challenges*”, i.e., how to combine more than one challenge of the above-mentioned 5-M challenges to model the even more practical problems to approximate the real-world applications. From the algorithm aspect, besides the above-mentioned five function-oriented approaches for solving complex problems, a promising way to design a more powerful EC algorithm is the “*Cooperation of different function-oriented approaches*”, which is also arrowed by a dashed line from the “Problem”. The arrow means that as the problems become more complex, they may also inspire more new function-oriented approaches that can be hybrid with the existing approaches introduced in this paper. Moreover, the “*Crossing of multi-disciplinary techniques*” is also a promising way to enhance EC algorithms to better solve complex continuous optimization problems. Nevertheless, the “Problem” and “Algorithm” themselves may have a relationship and can influence each other. Therefore, the “*Bi-directional interaction of problems and algorithms*” must be a significant future research topic. Also, the “*Balance of solution accuracy and computational burden*” is important for extending the EC algorithm in “*Applications of more real-world complex problems*” at last. These future research directions are summarized as the “3C-2B-1A” topics as illustrated in Fig. 10 and are discussed as follows.

4.1 Combination of 2-M or X-M challenges

Compared with the complex continuous optimization problems discussed above, real-world problems are often under an even more complex environment, which may involve 2-M or X-M challenges. For example, some application problems are not only large-scale with many dimensions but also dynamic with many changes, such as the large-scale dynamic community detection problem (Yin et al. 2021) and the large-scale dynamic optimal reactive power flow problem (Xiao et al. 2020). To solve these complex practical optimization problems, we need to build suitable models and use appropriate algorithms to optimize them. Therefore, one of the future research directions is how to make the problem model more practical by considering these X-M challenges existing in these problems. If we model the real-world application problems by only considering some easy characteristics, the problem model may not be suitable for practical use. Thus, it is a worthy research direction to consider different kinds of problem characteristics via multi-source data association and real data to combine different kinds of challenges (Liang et al. 2020; Shi et al. 2021; Wu et al. 2021a). This way, we will obtain much more complex optimization problems such as large-scale dynamic optimization problems, constrained multi-modal optimization problems, expensive constrained optimization problems, and dynamic constrained many-objective optimization problems, etc. Furthermore, extending EC algorithms to solve these more complex problems is also in great need.

Nevertheless, the different challenges may also share some common characteristics. For example, as shown in Fig. 1 and Table 1, both the MMOP and MOP have the many-optima challenge. Therefore, the relationship of MMOP and MOP can be a very interesting research topic to find out the bridge connecting multi-objective and multimodal (Chen et al. 2020b). This way, the different challenges can not only be combined but also be transformed, so that the EC algorithms designed for dealing with one challenge can also be adopted for dealing with another challenge.

4.2 Cooperation of different function-oriented approaches

In this paper, five function-oriented approaches are discussed for different complex continuous optimization problems. Each approach attempts to improve the performance of the algorithm from a different angle. If we hybrid these approaches, then ideally, the resulting cooperative approach can better solve complex continuous optimization problems, because it considers different aspects in dealing with the optimization problems or enhancing the EC algorithms. Moreover, as mentioned above, under the difficulty and challenges of 2-M or X-M, the cooperation of different function-oriented approaches will be more promising. For example, a reducing problem difficult approach can make the problem easier to be tackled by EC algorithms and an increasing algorithm diversity approach and/or an accelerating convergence speed approach will certainly further make the EC algorithm work better. For a problem, in an ideal situation, a reducing problem difficult approach is adopted to make the problem easy to solve, and then an increasing algorithm diversity approach and an accelerating convergence speed approach are adopted to improve the performance. Nevertheless, a simple hybrid of different function-oriented approaches may not work well. When adopting different function-oriented approaches to deal with a problem, these approaches should cooperate and promote with each other, not just the simple combination. Thus, how to efficiently control their cooperation is a worthy future research direction for EC algorithms. Moreover, different approaches may be required in different stages of the process in solving problems. Therefore, the adaptation of these function-oriented approaches is also worthy studied in the future. For example, the increasing diversity approach may be better to enhance the exploration ability of the algorithm and therefore should be emphasized in the early stage, while the accelerating convergence speed approach should be emphasized in the late or convergence stage to enhance the exploitation ability (Zhan et al. 2009, 2020). This way, a trade-off can be achieved between the exploration ability and the exploitation ability to better solve complex continuous optimization problems. Very recently, the new pipeline-based parallel technique for EC (Li et al. 2020b) and matrix-based EC (MEC) (Zhan et al. 2021) have been proposed, and are worthy for future study in solving complex continuous optimization problems, especially the LSOP.

4.3 Crossing of multi-disciplinary techniques

Besides the above cooperation of different function-oriented approaches for efficiently dealing with the complex continuous optimization problems, integrating ideas and techniques from the other disciplines into EC algorithms also has great potential to improve the performance of algorithms. For example, we can broaden our horizons and pay deeper attention to disciplines like biology and physics, to seek inspirations for enhancing the performance of existing EC algorithms, such as not only making strong scientific, mathematical, and/or statistical foundations to enhance the EC algorithms, but also providing solid and formal insights on the search behavior of the EC algorithms on the search space of solutions (Sorensen 2015). For the crossing of EC and biology, many EC algorithms, such as GA, ACO, and PSO, are inspired by natural phenomena. The recent advances in biology, particularly virology and immunology, may bring more inspiration to enhance the performance of existing EC algorithms. For the crossing of EC and physics, particle physics is a subject that can be focused on.

Artificial intelligence (AI) is also a research hotspot in recent years. Although the EC itself is a kind of powerful AI technique, some kinds of other famous AI techniques, such as artificial neural networks, are able to integrate with EC algorithms to improve the performance in solving complex continuous optimization problems. The feedback network and the ability of self-study and self-adaptability of the neural network provide another way for the parameter adjustment, the prediction for the new environments, the fitness imitation, and so on. Moreover, transfer learning in AI can inspire evolutionary transfer optimization, being a new frontier in EC research for complex optimization (Tan et al. 2021). Therefore, the crossing of inspirations and techniques between the multi-disciplines and EC algorithms is a promising way to better solve complex continuous optimization problems.

4.4 Bi-directional interaction of problems and algorithms

In the future, problems become more and more complex, while algorithms become stronger and stronger. In fact, the EC algorithm itself is also a complex system. So how to efficiently optimize this complex system (i.e., the EC algorithm) to efficiently solve other problems (e.g., the solved complex optimization problem) seems to be interesting. Referring to the defined 5-M challenges in complex continuous optimization problems, can we think that some or all of such challenges (or named characteristics) also exist in EC algorithms? For example, if we design an EC algorithm with very large population size, is the algorithm a large-scale system? In such a case, can we use the approaches that deal with large-scale challenges to enhance the performance of the large-scale population size EC algorithms? If we change the parameters or operators of an EC algorithm dynamically during the evolution, is the algorithm a dynamic system? In this case, the dynamic optimization techniques that deal with DOP may be useful for the enhancement of the dynamic EC algorithms. Therefore, the complexity and challenges of the problem and algorithm may have a relationship and interaction. How can complex problems interact with complex algorithms and vice versa? This bi-directional interaction of problems and algorithms will certainly be an interesting future research topic.

4.5 Balance of solution accuracy and computational burden

Moreover, as the increasing complexity of problems and algorithms, a necessary consideration is the balance of solution accuracy and computational burden, especially in those real-time practical applications. When we need to obtain a solution with higher accuracy through EC algorithms, it often requires more computational burden. However, the too heavy computational burden is not acceptable sometimes. So it is in need to obtain a relatively satisfactory solution within the available computational burden. To deal with the trade-off between solution accuracy and computational burden, there are two potential research directions. The first one is the analysis of "accuracy efficiency" for EC algorithms. In detail, accuracy efficiency represents the computational burden required to refine the solution accuracy to a certain degree. The analysis is about how to improve the computational accuracy and efficiency. The second one is the deployment of high performance computing and distributed computing resources, particularly supercomputing and cloud computing, because they provide powerful computational ability. Thus, the distributed EC (DEC) algorithms design (Zhan et al. 2017; Zhan et al. 2020) and the deployment of DEC algorithms on supercomputing or cloud computing platform are promising approaches to ease the computational burden.

4.6 Applications of more real-world complex problems

As surveyed in Sect. 3, many researchers have attempted to use EC algorithms for solving real-world complex continuous optimization problems and obtained some promising results. Therefore, at last but not least, with the improvement of EC algorithms in dealing with the 5-M and 4-V challenges in complex continuous optimization problems, EC algorithms have great potential to be extended to solve more complex continuous optimization problems in the real world, such as the complex problems in economy and society, in science and engineering, in industry and manufacturing, in business and finance, and in many fields that exist optimization problems. This is absolutely a hot and long-existing research topic in not only the EC and complex system communities but also many other areas that face complex continuous optimization problems.

5 Conclusion

In this paper, we have provided a comprehensive survey on the application of EC algorithms in solving complex continuous optimization problems. Taxonomy of research in this field is defined by the function-oriented approaches according to how the EC algorithms are enabled and enhanced to efficiently deal with the 5-M challenges of complex continuous optimization problems. The survey consists of EC algorithms for solving six types of complex continuous optimization problems, including LSOP, DOP, MMOP, MOP/MaOP, COP, and EOP. In each problem type, the related works are classified according to some or all of the five function-oriented approaches, including *reducing problem difficulty*, *increasing algorithm diversity*, *accelerating convergence speed*, *reducing running time*, and *extending application field*. Such a systematic and structural taxonomy provides a better understanding of how to efficiently use EC algorithms to well tackle complex continuous optimization problems. Based on the survey of the existing works in the literature, the 3C-2B-1A potential future research directions in this field have also been proposed and discussed. We hope this survey is able to well-organize existing works, elicit escalating attention, inspire new ideas, and extend wider and deeper research in this emerging, exciting, and interesting field.

Acknowledgements This work was supported in part by the National Key Research and Development Program of China under Grant 2019YFB2102102, in part by the Outstanding Youth Science Foundation under Grant 61822602, in part by the National Natural Science Foundations of China (NSFC) under Grant 61772207 and Grant 61873097, in part by the Key-Area Research and Development of Guangdong Province under Grant 2020B010166002, in part by the Guangdong Natural Science Foundation Research Team under Grant 2018B030312003, and in part by the Guangdong-Hong Kong Joint Innovation Platform under Grant 2018B050502006. (Corresponding author: Zhi-Hui Zhan).

References

Abdulkarim SA, Engelbrecht AP (2021) Time series forecasting with feedforward neural networks trained using particle swarm optimizers for dynamic environments. Neural Comput Appl 33:2667–2683

Adra SF, Fleming PJ (2011) Diversity management in evolutionary many-objective optimization. IEEE Trans Evol Comput 15:183–195

Akinsolu MO, Liu B, Grout V, Lazaridis PI, Mognaschi ME, Barba PD (2019) A parallel surrogate model assisted evolutionary algorithm for electromagnetic design optimization. IEEE Trans Emerg Topics Comput Intell 3:93–105

Ao Y (2012) Differential evolution using second mutation for high-dimensional real-parameter optimization. In: Proceedings of Measuring Technology and Mechatronics Automation in Electrical Engineering, New York, NY. Springer New York, pp. 191–201

Asafuddoula M, Ray T, Sarker R (2015) A decomposition-based evolutionary algorithm for many objective optimization. IEEE Trans Evol Comput 19:445–460

Asafuddoula M, Verma B, Zhang MJ (2018) A divide-and-conquer-based ensemble classifier learning by means of many-objective optimization. IEEE Trans Evol Comput 22:762–777

Bader J, Zitzler E (2011) Hype: an algorithm for fast hypervolume-based many-objective optimization. Evol Comput 19:45–76

Bandaru S, Deb K (2013) A parameterless-niching-assisted bi-objective approach to multimodal optimization. In: Proceedings of IEEE Congress on Evolutionary Computation. IEEE, pp 95–102

Basak A, Das S, Tan KC (2013) Multimodal optimization using a biobjective differential evolution algorithm enhanced with mean distance-based selection. IEEE Trans Evol Comput 17:666–685

Biswas S, Kundu S, Das S (2014) An improved parent-centric mutation with normalized neighborhoods for inducing niching behavior in differential evolution. IEEE Trans Cybern 44:1726–1737

Biswas S, Kundu S, Das S (2015) Inducing niching behavior in differential evolution through local information sharing. IEEE Trans Evol Comput 19:246–263

Bitsi K, Wallmark O, Bosga S (2019) Many-objective optimization of ipm and induction motors for automotive application. In: Proceedings of 21st European Conference on Power Electronics and Applications. pp. 1–10

Bonabeau E, Dorigo M, Theraulaz G (2000) Inspiration for optimization from social insect behaviour. Nature 406:39–42

Branke J, Kaussler T, Smidt C, Schmeck H (2000) A multi-population approach to dynamic optimization problems. In: Evolutionary Design and Manufacture. pp. 299–307

Brest J, Maucec MS (2011) Self-adaptive differential evolution algorithm using population size reduction and three strategies. Soft Comput 15:2157–2174

Brest J, Greiner S, Boskovic B, Mernik M, Zumer V (2006) Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems. IEEE Trans Evol Comput 10:646–657

Brest J, Boskovic B, Zamuda A, Fister I, Maucec MS (2012) Self-adaptive differential evolution algorithm with a small and varying population size. In: Proceedings of IEEE Congress on Evolutionary Computation. IEEE, pp. 1–8

Buche D, Schraudolph NN, Koumoutsakos P (2005) Accelerating evolutionary algorithms with gaussian process fitness function models. IEEE Trans Syst Man Cyern Part C-Appl Rev 35:183–194

Cai XY, Yang ZX, Fan Z, Zhang QF (2017) Decomposition-based-sorting and angle-based-selection for evolutionary multiobjective and many-objective optimization. IEEE Trans Cybern 47:2824–2837

Campos Ad, Pozo ATR, Duarte EP (2019) Parallel multi-swarm PSO strategies for solving many objective optimization problems. J Parallel Distrib Comput 126:13–33

Cao L, Xu L, Goodman ED (2019) A collaboration-based particle swarm optimizer with history-guided estimation for optimization in dynamic environments. Expert Syst Appl 120:1–13

Chen CH, Chou JH (2017) Multiobjective optimization of airline crew roster recovery problems under disruption conditions. IEEE Trans Syst Man Cybern Syst 47:133–144

Chen JH, Goldberg D, Ho SY, Sastry K (2002) Fitness inheritance in multi-objective optimization. In: Proceedings of Genetic and Evolutionary Computation Conference. pp. 319–326

Chen ZG, Du KL, Zhan ZH, Zhang L (2015a) Deadline constrained cloud computing resources scheduling for cost optimization based on dynamic objective genetic algorithm. In: Proceedings of IEEE Congress on Evolutionary Computation. pp. 708–714

Chen ZG et al. (2015b) Deadline constrained cloud computing resources scheduling through an ant colony system approach. In: Proceedings of International Conference on Cloud Computing Research and Innovation. pp. 112–119

Chen ZG et al (2019) Multiobjective cloud workflow scheduling: a multiple populations ant colony system approach. IEEE Trans Cybern 49:2912–2926

Chen ZG, Zhan ZH, Wang H, Zhang J (2020a) Distributed individuals for multiple peaks: a novel differential evolution for multimodal optimization problems. IEEE Trans Evol Comput 24:708–719

Chen ZG, Zhan ZH, Zhang J (2020b) Bridge connecting multiobjetive and multimodal: a new approach for multiobjetive optimization via multimodal optimization. In: Proc IEEE Int Conf Inf, Cybern, and Comput Social Syst. pp. 463–468

Cheng R, Jin YC (2015a) A competitive swarm optimizer for large scale optimization. IEEE Trans Cybern 45:191–204

Cheng R, Jin YC (2015b) A social learning particle swarm optimization algorithm for scalable optimization. Inf Sci 291:43–60

Cheng JX, Zhang GX, Neri F (2013) Enhancing distributed differential evolution with multicultural migration for global numerical optimization. Inf Sci 247:72–93

Cheng R, Jin YC, Olhofer M, Sendhoff B (2016) A reference vector guided evolutionary algorithm for many-objective optimization. IEEE Trans Evol Comput 20:773–791

Cheng R, Rodemann T, Fischer M, Olhofer M, Jin Y (2017) Evolutionary many-objective optimization of hybrid electric vehicle control: from general optimization to preference articulation. IEEE Trans Emerg Topics Comput Intell 1:97–111

Cheng R, Li MQ, Li K, Yao X (2018) Evolutionary multiobjective optimization-based multimodal optimization: fitness landscape approximation and peak detection. IEEE Trans Evol Comput 22:692–706

Cheung YM, Gu FQ, Liu HL (2016) Objective extraction for many-objective optimization problems: algorithm and test problems. IEEE Trans Evol Comput 20:755–772

Chugh T, Sindhya K, Miettinen K, Jin YC, Kratky T, Makkonen P (2017) Surrogate-assisted evolutionary multiobjective shape optimization of an air intake ventilation system. In: Proceedings of IEEE Congress on Evolutionary Computation. pp. 1541–1548

Chugh T, Jin YC, Miettinen K, Hakanen J, Sindhya K (2018) A surrogate-assisted reference vector guided evolutionary algorithm for computationally expensive many-objective optimization. IEEE Trans Evol Comput 22:129–142

Das S, Mandal A, Mukherjee R (2014) An adaptive differential evolution algorithm for global optimization in dynamic environments. IEEE Trans Cybern 44:966–978

Deb K (2000) An efficient constraint handling method for genetic algorithms. Comput Meth Appl Mech Eng 186:311–338

Deb K, Jain H (2014) An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: solving problems with box constraints. IEEE Trans Evol Comput 18:577–601

Deb K, Saha A (2012) Multimodal optimization using a bi-objective evolutionary algorithm. Evol Comput 20:27–62

Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: Nsga-II. IEEE Trans Evol Comput 6:182–197

Deng J, Zhang Q (2019) Approximating hypervolume and hypervolume contributions using polar coordinate. IEEE Trans Evol Comput 23:913–918

Deng H, Peng L, Zhang H, Yang B, Chen Z (2019) Ranking-based biased learning swarm optimizer for large-scale optimization. Inf Sci 493:120–137

Depolli M, Trobec R, Filipič B (2013) Asynchronous master-slave parallelization of differential evolution for multi-objective optimization. Evol Comput 21:261–291

Du W, Li B (2008) Multi-strategy ensemble particle swarm optimization for dynamic optimization. Inf Sci 178:3096–3109

Eiben AE, Smith J (2015) From evolutionary computation to the evolution of things. Nature 521:476–482

Fan Z et al (2019) An improved epsilon constraint-handling method in moea/d for cmops with large infeasible regions. Soft Comput 23:12491–12510

Fleck M, Troya J, Kessentini M, Wimmer M, Alkhazi B (2017) Model transformation modularization as a many-objective optimization problem. IEEE Trans Softw Eng 43:1009–1032

Fogel DB (1995) Evolutionary computation: Toward a new philosophy of machine intelligence. IEEE

Frey BJ, Dueck D (2007) Clustering by passing messages between data points. Science 315:972–976

Gao WF, Yen GG, Liu SY (2014) A cluster-based differential evolution with self-adaptive strategy for multimodal optimization. IEEE Trans Cybern 44:1314–1327

Gao WF, Yen GG, Liu SY (2015) A dual-population differential evolution with coevolution for constrained optimization. IEEE Trans Cybern 45:1094–1107

Ge YF, Yu WJ, Zhang J (2016) Diversity-based multi-population differential evolution for large-scale optimization. In: Proceedings of the 2016 on Genetic and Evolutionary Computation Conference Companion, Colorado, USA. pp. 31–32

Ge Y, Yu W, Zhan Z, Zhang J (2018a) Competition-based distributed differential evolution. In: Proceeding of IEEE Congress on Evolutionary Computation. pp. 1–8

Ge YF, Yu WJ, Lin Y, Gong YJ, Zhan ZH, Chen WN, Zhang J (2018b) Distributed differential evolution based on adaptive mergence and split for large-scale optimization. IEEE Trans Cybern 48:2166–2180

Goharrizi AY, Singh R, Gole AM, Filizadeh S, Muller JC, Jayasinghe RP (2015) A parallel multimodal optimization algorithm for simulation-based design of power systems. IEEE Trans Power Deliv 30:2128–2137

Gong WY, Cai ZH, Liang DW (2015a) Adaptive ranking mutation operator based differential evolution for constrained optimization. IEEE Trans Cybern 45:716–727

Gong YJ, Chen WN, Zhan ZH, Zhang J, Li Y, Zhang QF, Li JJ (2015b) Distributed evolutionary algorithms and their models: A survey of the state-of-the-art. Appl Soft Comput 34:286–300

Gong WY, Wang Y, Cai ZH, Yang SX (2017) A weighted biobjective transformation technique for locating multiple optimal solutions of nonlinear equation systems. IEEE Trans Evol Comput 21:697–713

Guerrero-Mosquera C, Verleysen M, Vazquez AN (2011) Dimensionality reduction for eeg classification using mutual information and svm. In: Proceedings of IEEE International Workshop on Machine Learning for Signal Processing. pp. 1–6.

Guo D, Chai TY, Ding JL, Jin YC (2016) Small data driven evolutionary multi-objective optimization of fused magnesium furnaces. In: Proceedings of IEEE Symposium Series on Computational Intelligence. pp. 1–8

Guo W, Si C, Xue Y, Mao Y, Wang L, Wu Q (2018) A grouping particle swarm optimizer with personal-best-position guidance for large scale optimization. IEEE/ACM Trans Comput Biol Bioinf 15:1904–1915

Guo YN, Zhang X, Gong DW, Zhang Z, Yang JJ (2020) Novel interactive preference-based multiobjective evolutionary optimization for bolt supporting networks. IEEE Trans Evol Comput 24:750–764

Halder U, Das S, Maity D (2013) A cluster-based differential evolution algorithm with external archive for optimization in dynamic environments. IEEE Trans Cybern 43:881–897

Hamida SB, Schoenauer M (2002) Aschea: New results using adaptive segregational constraint handling. In: Proceedings of IEEE Congress on Evolutionary Computation. IEEE, pp. 884–889

Hamza NM, Essam DL, Sarker RA (2016) Constraint consensus mutation-based differential evolution for constrained optimization. IEEE Trans Evol Comput 20:447–459

Hashemi AB, Meybodi MR (2009) Cellular PSO: A PSO for dynamic environments. In: Proceedings of International Symposium on Intelligence Computation and Applications. pp. 422–433

He ZN, Yen GG, Zhang J (2014) Fuzzy-based pareto optimality for many-objective evolutionary algorithms. IEEE Trans Evol Comput 18:269–285

He XY, Zhou YR, Chen ZF, Zhang QF (2019) Evolutionary many-objective optimization based on dynamical decomposition. IEEE Trans Evol Comput 23:361–375

Hellwig M, Beyer H (2018) A matrix adaptation evolution strategy for constrained real-parameter optimization. In: 2018 IEEE Congress on Evolutionary Computation (CEC), 8–13 July 2018. pp. 1–8

Hsieh ST, Chiu SY, Yen SJ (2012) Adoptive population differential evolution with local search for solving large scale global optimization. In: Proceedings of IEEE International Conference on Systems, Man, and Cybernetics. IEEE, pp. 1090–1094

Hsieh YC, Lee YC, You PS (2015) Solving nonlinear constrained optimization problems: An immune evolutionary based two-phase approach. Appl Math Model 39:5759–5768

Hu W, Yen GG, Luo GC (2017) Many-objective particle swarm optimization using two-stage strategy and parallel cell coordinate system. IEEE Trans Cybern 47:1446–1459

Hui S, Suganthan PN (2016) Ensemble and arithmetic recombination-based speciation differential evolution for multimodal optimization. IEEE Trans Cybern 46:64–74

Irawan D, Naujoks B, Emmerich M (2020) Cooperative-coevolution-cma-es with two-stage grouping. In: Proceeding of IEEE Congress on Evolutionary Computation. pp. 1–8

Ishibuchi H, Sakane Y, Tsukamoto N, Nojima Y (2009) Evolutionary many-objective optimization by nsga-ii and moea/d with large populations. In: Proceedings of IEEE International Conference on Systems, Man and Cybernetics. Pp. 1758–1763

Ishibuchi H, Sakane Y, Tsukamoto N, Nojima Y (2010) Simultaneous use of different scalarizing functions in moea/d. In: Proceedings of Genetic and Evolutionary Computation Conference. pp. 519–526

Jian JR, Zhan ZH, Zhang J (2020) Large-scale evolutionary optimization: A survey and experimental comparative study. Int J Mach Learn Cybern 11:729–745

Jian JR, Chen ZG, Zhan ZH, Zhang J (2021) Region encoding helps evolutionary computation evolve faster: A new solution encoding scheme in particle swarm for large-scale optimization. IEEE Trans Evolut Comput. https://doi.org/10.1109/TEVC.2021.3065659

Jiang SY, Yang SX (2017) A strength pareto evolutionary algorithm based on reference direction for multiobjective and many-objective optimization. IEEE Trans Evol Comput 21:329–346

Jiang S, Zhang J, Ong Y, Zhang AN, Tan PS (2015) A simple and fast hypervolume indicator-based multiobjective evolutionary algorithm. IEEE Trans Cybern 45:2202–2213

Jin Y (2005) A comprehensive survey of fitness approximation in evolutionary computation. Soft Comput 9:3–12

Jin YC (2011) Surrogate-assisted evolutionary computation: Recent advances and future challenges. Swarm Evol Comput 1:61–70

Jin YC, Sendhoff B (2004) Reducing fitness evaluations using clustering techniques and neural network ensembles. In: Proceedings of Genetic and Evolutionary Computation. pp. 688–699

Jin YC, Sendhoff B (2009) A systems approach to evolutionary multiobjective structural optimization and beyond. IEEE Comput Intell Mag 4:62–76

Jin YC, Olhofer M, Sendhoff B (2002) A framework for evolutionary optimization with approximate fitness functions. IEEE Trans Evol Comput 6:481–494

Jin YC, Wang HD, Chugh T, Guo D, Miettinen K (2019) Data-driven evolutionary optimization: An overview and case studies. IEEE Trans Evol Comput 23:442–458

Karakasis MK, Giotis AP, Giannakoglou KC (2003) Inexact information aided low-cost distributed genetic algorithms for aerodynamic shape optimization. Int J Numer Methods Fluids 43:1149–1166

Karakasis MK, Koubogiannis DG, Giannakoglou KC (2007) Hierarchical distributed metamodel-assisted evolutionary algorithms in shape optimization. Int J Numer Methods Fluids 53:455–469

Kazimipour B, Omidvar MN, Qin AK, Li XD, Yao X (2019) Bandit-based cooperative coevolution for tackling contribution imbalance in large-scale optimization problems. Appl Soft Comput 76:265–281

Kim HS, Cho SB (2001) An efficient genetic algorithm with less fitness evaluation by clustering. In: Proceedings of Congress on Evolutionary Computation. IEEE, pp. 887–894

Kordestani JK, Ranginkaman AE, Meybodi MR, Novoa-Hernandez P (2019) A novel framework for improving multi-population algorithms for dynamic optimization problems: A scheduling approach. Swarm Evol Comput 44:788–805

Kovaleva M, Bulger D, Zeb BA, Esselle KP (2017) Cross-entropy method for electromagnetic optimization with constraints and mixed variables. IEEE Trans Antennas Propag 65:5532–5540

Koziel S, Michalewicz Z (1999) Evolutionary algorithms, homomorphous mappings, and constrained parameter optimization. Evol Comput 7:19–44

Kushida J, Hara A, Takahama T (2015) Rank-based differential evolution with multiple mutation strategies for large scale global optimization. In: Proceedings of IEEE Congress on Evolutionary Computation. IEEE, pp. 353–360

Lan R, Zhu Y, Lu H, Liu Z, Luo X (2020) A two-phase learning-based swarm optimizer for large-scale optimization. IEEE Trans Cybern. https://doi.org/10.1109/TCYB.2020.2968400

LaTorre A, Muelas S, Pena JM (2012) Multiple offspring sampling in large scale global optimization. In: Proceedings of IEEE Congress on Evolutionary Computation. IEEE, pp. 1–8

Li XD (2005) Efficient differential evolution using speciation for multimodal function optimization. In: Proceedings of Genetic and Evolutionary Computation Conference, Washington, DC. pp. 873–880

Li LX, Tang K (2015) History-based topological speciation for multimodal optimization. IEEE Trans Evol Comput 19:136–150

Li CH, Yang SX (2008) Fast multi-swarm optimization for dynamic optimization problems. In: Proceedings of 4th International Conference on Natural Computation. pp. 624–628

Li CH, Yang SX (2012) A general framework of multipopulation methods with clustering in undetectable dynamic environments. IEEE Trans Evol Comput 16:556–577

Li XD, Yao X (2009) Tackling high dimensional nonseparable optimization problems by cooperatively coevolving particle swarms. In: Proceedings of IEEE Congress on Evolutionary Computation. IEEE, pp 1546–1553

Li XD, Yao X (2012) Cooperatively coevolving particle swarms for large scale optimization. IEEE Trans Evol Comput 16:210–224

Li JP, Balazs ME, Parks GT, Clarkson PJ (2002) A species conserving genetic algorithm for multimodal function optimization. Evol Comput 10:207–234

Li MQ, Yang SX, Liu XH (2014a) Shift-based density estimation for pareto-based algorithms in many-objective optimization. IEEE Trans Evol Comput 18:348–365

Li YZ, Wu QH, Li MS, Zhan JP (2014b) Mean-variance model for power system economic dispatch with wind power integrated. Energy 72:510–520

Li J, Chen W, Zhang J, Zhan Z (2015a) A parallel implementation of multiobjective particle swarm optimization algorithm based on decomposition. In: Proceedings of IEEE Symposium Series on Computational Intelligence. pp. 1310–1317

Li K, Deb K, Zhang QF, Kwong S (2015b) An evolutionary many-objective optimization algorithm based on dominance and decomposition. IEEE Trans Evol Comput 19:694–716

Li BD, Tang K, Li JL, Yao X (2016a) Stochastic ranking algorithm for many-objective optimization based on multiple indicators. IEEE Trans Evol Comput 20:924–938

Li CH, Nguyen TT, Yang M, Mavrovouniotis M, Yang SX (2016b) An adaptive multipopulation framework for locating and tracking multiple optima. IEEE Trans Evol Comput 20:590–605

Li EY, Wang H, Ye F (2016c) Two-level multi-surrogate assisted optimization method for high dimensional nonlinear problems. Appl Soft Comput 46:26–36

Li H, Zhang Q, Deng J (2017) Biased multiobjective optimization and decomposition algorithm. IEEE Trans Cybern 47:52–66

Li L, Fang W, Wang Q, Sun J (2019a) Differential grouping with spectral clustering for large scale global optimization. In: Proceedings of IEEE Congress on Evolutionary Computation. pp. 334–341

Li Y, Zhan Z, Jin H, Zhang J (2019b) Cloudde-based distributed differential evolution for solving dynamic optimization problems. In: Proceedings of Tenth International Conference on Intelligent Control and Information Processing. IEEE, pp. 94–99

Li J, Li J, Pardalos PM, Yang C (2020a) Dmaoea-εc: Decomposition-based many-objective evolutionary algorithm with the ε-constraint framework. Inf Sci 537:203–226

Li JY, Zhan ZH, Liu RD, Wang C, Kwong S, Zhang J (2020b) Generation-level parallelism for evolutionary computation: A pipeline-based parallel particle swarm optimization. IEEE Trans Cybern, 1–12

Li JY, Zhan ZH, Wang C, Jin H, Zhang J (2020c) Boosting data-driven evolutionary algorithm with localized data generation. IEEE Trans Evol Comput 24:923–937

Li JY, Zhan ZH, Wang H, Zhang J (2020d) Data-driven evolutionary algorithm with perturbation-based ensemble surrogates. IEEE Trans Cybern, 1–13

Li D, Guo W, Lerch A, Li Y, Wang L, Wu Q (2021) An adaptive particle swarm optimizer with decoupled exploration and exploitation for large scale optimization. Swarm Evol Comput 60:100789

Liang D, Zhan ZH, Zhang YC, Zhang J (2020) An efficient ant colony system approach for new energy vehicle dispatch problem. IEEE Trans Intell Transp Syst 21:4784–4797

Lim D, Jin YC, Ong YS, Sendhoff B (2010) Generalizing surrogate-assisted evolutionary computation. IEEE Trans Evol Comput 14:329–355

Ling YB, Li HJ, Cao B (2016) Cooperative co-evolution with graph-based differential grouping for large scale global optimization. In: Proceedings of 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery. IEEE, pp. 95–102

Liu JP, Tang K (2013) Scaling up covariance matrix adaptation evolution strategy using cooperative coevolution. In: Proceedings of 14th International Conference on Intelligent Data Engineering and Automated Learning, Hefei, CHINA. pp. 350–357

Liu Y, Yao X, Zhao QF, Higuchi T (2001) Scaling up fast evolutionary programming with cooperative coevolution. In: Proceedings of the 2001 Congress on Evolutionary Computation, Vols 1 and 2. pp. 1101–1108

Liu LL, Yang SX, Wang DW (2010) Particle swarm optimization with composite particles in dynamic environments. IEEE Trans Syst Man Cybern Part B-Cybern 40:1634–1648

Liu LL, Wang DW, Tang JF (2011) Composite particle optimization with hyper-reflection scheme in dynamic environments. Appl Soft Comput 11:4626–4639

Liu JJ, Teo KL, Wang XY, Wu CZ (2016a) An exact penalty function-based differential search algorithm for constrained global optimization. Soft Comput 20:1305–1313

Liu XF, Zhan ZH, Lin JH, Zhang J (2016b) Parallel differential evolution based on distributed cloud computing resources for power electronic circuit optimization. In: Proceedings of Genetic and Evolutionary Computation Conference. pp. 117–118

Liu XF, Zhan ZH, Zhang J (2017) An energy aware unified ant colony system for dynamic virtual machine placement in cloud computing. Energies 10:609

Liu F, Sun Y, Wang G, Wu T (2018a) An artificial bee colony algorithm based on dynamic penalty and lévy flight for constrained optimization problems. Arab J Sci Eng 43:7189–7208

Liu XF, Zhan ZH, Deng JD, Li Y, Gu TL, Zhang J (2018b) An energy efficient ant colony system for virtual machine placement in cloud computing. IEEE Trans Evol Comput 22:113–128

Liu XF, Zhan ZH, Zhang J (2018c) Neural network for change direction prediction in dynamic optimization. IEEE Access 6:72649–72662

Liu RD, Chen ZG, Wang ZJ, Zhan ZH (2019a) Intelligent path planning for auvs in dynamic environments: An eda-based learning fixed height histogram approach. IEEE Access 7:185433–185446

Liu W, Zhou Y, Li B, Tang K (2019b) Cooperative co-evolution with soft grouping for large scale global optimization. In: Proceedings of IEEE Congress on Evolutionary Computation. pp. 318–325

Liu XF, Zhan ZH, Gao Y, Zhang J, Kwong S, Zhang J (2019c) Coevolutionary particle swarm optimization with bottleneck objective learning strategy for many-objective optimization. IEEE Trans Evol Comput 23:587–602

Liu XF, Zhou YR, Yu X, Lin Y (2019d) Dual-archive-based particle swarm optimization for dynamic optimization. Appl Soft Comput 85:105876

Liu XF, Zhan ZH, Gu TL, Kwong S, Lu ZY, Duh HBL, Zhang J (2020) Neural network-based information transfer for dynamic optimization. IEEE Trans Neural Netw Learn Syst 31:1557–1570

Liu SC, Chen C, Zhan ZH, Zhang J (2021a) Multi-objective emergency resource dispatch based on coevolutionary multiswarm particle swarm optimization. In: Proceedings of International Conference on Evolutionary Multi-Criterion Optimization. pp. 746–758

Liu SC, Zhan ZH, Tan KC, Zhang J (2021b) A multi-objective framework for many-objective optimization. IEEE Trans Cybern. https://doi.org/10.1109/TCYB.2021.3082200

Liu XF, Zhan ZH, Zhang J (2021c) Resource-aware distributed differential evolution for training expensive neural-network-based controller in power electronic circuit. IEEE Trans Neural Networks Learn Syst. https://doi.org/10.1109/TNNLS.2021.3075205

Luna F, Alba E (2015) Parallel multiobjective evolutionary algorithms. In: Kacprzyk J, Pedrycz W (eds) Springer handbook of computational intelligence. Springer, pp 1017–1031. https://doi.org/10.1007/978-3-662-43505-2_50

Luo X, Wang Z, Guan R, Zhan Z, Gao Y (2019) A distributed multiple populations framework for evolutionary algorithm in solving dynamic optimization problems. IEEE Access 7:44372–44390

Ma N, Liu X, Zhan Z, Zhong J, Zhang J (2017) Load balance aware distributed differential evolution for computationally expensive optimization problems. In: Proceedings of Genetic and Evolutionary Computation Conference. pp. 209–210

Ma LJ, Li JQ, Lin QZ, Gong MG, Coello CAC, Ming Z (2019) Reliable link inference for network data with community structures. IEEE Trans Cybern 49:3347–3361

Maesani A, Iacca G, Floreano D (2016) Memetic viability evolution for constrained optimization. IEEE Trans Evol Comput 20:125–144

Mahmoudzadeh S, Powers DMW, Atyabi A (2019) Uuv's hierarchical DE-based motion planning in a semi dynamic underwater wireless sensor network. IEEE Trans Cybern 49:2992–3005

Mallipeddi R, Suganthan PN (2010) Ensemble of constraint handling techniques. IEEE Trans Evol Comput 14:561–579

Mane S, Rao M (2017) Many-objective optimization: Problems and evolutionary algorithms–a short review. Int J Appl Eng Res 12:9774–9793

Martinez SZ, Coello CAC (2013) Moea/d assisted by rbf networks for expensive multi-objective optimization problems. In: Proceedings of Genetic and Evolutionary Computation Conference. pp. 1405–1412

Martins R, Lourenco N, Horta N, Yin J, Mak PI, Martins RP (2019) Many-objective sizing optimization of a class-c/d vco for ultralow-power iot and ultralow-phase-noise cellular applications. IEEE Trans Very Large Scale Integration (VLSI) Syst, 27: 69–82

Matos JL, Britto A (2017) Multi-swarm algorithm based on archiving and topologies for many-objective optimization. In: Proceedings of IEEE Congress on Evolutionary Computation. pp. 1877–1884

Mei Y, Li XD, Yao X (2014) Cooperative coevolution with route distance grouping for large-scale capacitated arc routing problems. IEEE Trans Evol Comput 18:435–449

Mei Y, Omidvar MN, Li XD, Yao X (2016) A competitive divide-and-conquer algorithm for unconstrained large-scale black-box optimization. ACM Trans Math Softw 42:1–24

Miranda PBC, Morais RFAB, Silva RMA (2018) Using a many-objective optimization algorithm to select sampling approaches for imbalanced datasets. In: Proceedings of IEEE Congress on Evolutionary Computation. pp. 1–7

Molina D, Herrera F (2015) Iterative hybridization of DE with local search for the cec'2015 special session on large scale global optimization. In: Proceedings of IEEE Congress on Evolutionary Computation. IEEE, pp. 1974–1978

Molina D, Lozano M, Herrera F (2010) Ma-sw-chains: Memetic algorithm based on local search chains for large scale continuous global optimization. In: Proceedings of IEEE Congress on Evolutionary Computation. IEEE, pp. 1–8

Molina D, LaTorre A, Herrera F (2018) Shade with iterative local search for large-scale global optimization. In: Proceedings of IEEE Congress on Evolutionary Computation. IEEE, pp. 1252–1259

Müller N, Glasmachers T (2018) Challenges in high-dimensional reinforcement learning with evolution strategies. In, Cham. Parallel Problem Solving from Nature—PPSN XV. pp. 411–423

Naidu YR, Ojha AK (2018) Solving multiobjective optimization problems using hybrid cooperative invasive weed optimization with multiple populations. IEEE Trans Syst Man Cybern Syst 48:821–832

Nekooei SM, Chen G (2020) Cooperative coevolution design of multilevel fuzzy logic controllers for media access control in wireless body area networks. IEEE Trans Emerg Topics Comput Intell 4:336–350

Nguyen S, Zhang M, Tan KC (2017) Surrogate-assisted genetic programming with simplified models for automated design of dispatching rules. IEEE Trans Cybern 47:2951–2965

Nickabadi A, Ebadzadeh MM, Safabakhsh R (2012) A competitive clustering particle swarm optimizer for dynamic optimization problems. Swarm Intell 6:177–206

Noroozi V, Hashemi AB, Meybodi MR (2011) Cellularde: A cellular based differential evolution for dynamic optimization problems. In: Proceedings of International Conference on Adaptive and Natural Computing Algorithms. pp. 340–349

Omidvar MN, Li XD, Yao X (2010) Cooperative co-evolution with delta grouping for large scale non-separable function optimization. In: Proceedings of IEEE Congress on Evolutionary Computation. IEEE

Omidvar MN, Li XD, Yao X (2011) Smart use of computational resources based on contribution for cooperative co-evolutionary algorithms. In: Proceedings of the 13th Annual Genetic and Evolutionary Computation Conference. pp. 1115–1122

Omidvar MN, Li XD, Mei Y, Yao X (2014) Cooperative co-evolution with differential grouping for large scale optimization. IEEE Trans Evol Comput 18:378–393

Omidvar MN, Kazimipour B, Li XD, Yao X (2016) Cbcc3-a contribution-based cooperative co-evolutionary algorithm with improved exploration/exploitation balance. In: Proceedings of IEEE Congress on Evolutionary Computation. IEEE, pp. 3541–3548

Omidvar MN, Yang M, Mei Y, Li XD, Yao X (2017) Dg2: A faster and more accurate differential grouping for large-scale black-box optimization. IEEE Trans Evol Comput 21:929–942

Peng M, Li C (2020) A contribution-based resource allocation scheme for multi-population methods in dynamic environments. In: 2020 2nd International Conference on Industrial Artificial Intelligence

Peng X, Jin Y, Wang H (2019) Multimodal optimization enhanced cooperative coevolution for large-scale optimization. IEEE Trans Cybern 49:3507–3520

Potter MA, De Jong KA (1994) A cooperative coevolutionary approach to function optimization. Parallel Problem Solving from Nature, 249–257

Purshouse RC, Fleming PJ (2007) On the evolutionary optimization of many conflicting objectives. IEEE Trans Evol Comput 11:770–784

Qu BY, Suganthan PN, Liang JJ (2012) Differential evolution with neighborhood mutation for multimodal optimization. IEEE Trans Evol Comput 16:601–614

Qu BY, Suganthan PN, Das S (2013) A distance-based locally informed particle swarm model for multimodal optimization. IEEE Trans Evol Comput 17:387–402

Rakitianskaia A, Engelbrecht AP (2008) Cooperative charged particle swarm optimiser. In: Proceedings of IEEE Congress on Evolutionary Computation. IEEE, pp. 933–939

Regis RG (2014) Evolutionary programming for high-dimensional constrained expensive black-box optimization using radial basis functions. IEEE Trans Evol Comput 18:326–347

Ren Z, Liang Y, Zhang A, Yang Y, Feng Z, Wang L (2019) Boosting cooperative coevolution for large scale optimization with a fine-grained computation resource allocation strategy. IEEE Trans Cybern 49:4180–4193

Runarsson TP, Yao X (2000) Stochastic ranking for constrained evolutionary optimization. IEEE Trans Evol Comput 4:284–294

Sabioni CL, Ribeiro MFO, Vasconcelos JA (2018) Robust design of an axial-flux permanent magnet synchronous generator based on many-objective optimization approach. IEEE Trans Magn 54:1–4

Saha C, Das S, Pal K, Mukherjee S (2016) A fuzzy rule-based penalty function approach for constrained evolutionary optimization. IEEE Trans Cybern 46:2953–2965

Salami M, Hendtlass T (2003) A fast evaluation strategy for evolutionary algorithms. Appl Soft Comput 2:156–173

Salimi A, Lowther DA (2016) Projection-based objective space reduction for many-objective optimization problems: Application to an induction motor design. In: Proceedings of IEEE Conference on Electromagnetic Field Computation

Sastry K, Goldberg DE, Pelikan M (2001) Don’t evaluate, inherit. In: Proceedings of Genetic and Evolutionary Computation Conference. pp. 551–558

Saxena DK, Duro JA, Tiwari A, Deb K, Zhang QF (2013) Objective reduction in many-objective optimization: linear and nonlinear algorithms. IEEE Trans Evol Comput 17:77–99

Shan SQ, Wang GG (2010) Survey of modeling and optimization strategies to solve high-dimensional design problems with computationally-expensive black-box functions. Struct Multidiscip Optim 41:219–241

Shang K, Ishibuchi H (2020) A new hypervolume-based evolutionary algorithm for many-objective optimization. IEEE Trans Evol Comput 24:839–852

Sharifi A, Noroozi V, Bashiri M, Hashemi AB, Meybodi MR (2012) Two phased cellular PSO: A new collaborative cellular algorithm for optimization in dynamic environments. In: Proceedings of IEEE Congress on Evolutionary Computation. IEEE, pp. 1–8

Shi YJ, Teng HF, Li ZQ (2005) Cooperative co-evolutionary differential evolution for function optimization. In: Proceedings of 1st International Conference on Natural Computation, Changsha, CHINA. pp. 1080–1088

Shi L, Zhan ZH, Liang D, Zhang J (2021) Memory-based ant colony system approach for multi-source data associated dynamic electric vehicle dispatch optimization. IEEE Intelligent Transportation Systems Transactions

Singh HK, Isaacs A, Ray T (2011) A pareto corner search evolutionary algorithm and dimensionality reduction in many-objective optimization problems. IEEE Trans Evol Comput 15:539–556

Song W, Wang Y, Li HX, Cai ZX (2015) Locating multiple optimal solutions of nonlinear equation systems based on multiobjective optimization. IEEE Trans Evol Comput 19:414–431

Sorensen K (2015) Metaheuristics-the metaphor exposed. Int Trans Oper Res 22:3–18

Spettel P, Beyer HG, Hellwig M (2019) A covariance matrix self-adaptation evolution strategy for optimization under linear constraints. IEEE Trans Evol Comput 23:514–524

Srinivas N, Deb K (1994) Muiltiobjective optimization using nondominated sorting in genetic algorithms. Evol Comput 2:221–248

Starkey A, Hagras H, Shakya S, Owusu G (2019) Ipatch: A many-objective type-2 fuzzy logic system for field workforce optimization. IEEE Trans Fuzzy Syst 27:502–514

Stoean C, Preuss M, Stoean R, Dumitrescu D (2010) Multimodal optimization by means of a topological species conservation algorithm. IEEE Trans Evol Comput 14:842–864

Sun CL, Zeng JC, Pan JS, Xue SD, Jin YC (2013) A new fitness estimation strategy for particle swarm optimization. Inf Sci 221:355–370

Sun Y, Kirley M, Halgamuge SK (2015) Extended differential grouping for large scale global optimization with direct and indirect variable interactions. In: Proceedings of the 2015 Genetic and Evolutionary Computation Conference. pp. 313–320

Sun CL, Jin YC, Cheng R, Ding JL, Zeng JC (2017) Surrogate-assisted cooperative swarm optimization of high-dimensional expensive problems. IEEE Trans Evol Comput 21:644–660

Sun Y, Kirley M, Halgamuge SK (2018a) A recursive decomposition method for large scale continuous optimization. IEEE Trans Evol Comput 22:647–661

Sun Y, Omidvar MN, Kirley M, Li X (2018b) Adaptive threshold parameter estimation with recursive differential grouping for problem decomposition. In: Proceedings of the Genetic and Evolutionary Computation Conference, Kyoto, Japan. pp. 889–896

Sun Y, Li X, Ernst A, Omidvar MN (2019a) Decomposition for large-scale optimization problems with overlapping components. In: Proceedings of IEEE Congress on Evolutionary Computation. pp. 326–333

Sun YA, Yen GG, Yi Z (2019b) Igd indicator-based evolutionary algorithm for many-objective optimization problems. IEEE Trans Evol Comput 23:173–187

Sun YN, Xue B, Zhang MJ, Yen GG (2019c) A new two-stage evolutionary algorithm for many-objective optimization. IEEE Trans Evol Comput 23:748–761

Sun YZ, Wang J, Lu ZH (2019d) Asynchronous parallel surrogate optimization algorithm based on ensemble surrogating model and stochastic response surface method application to quantitative strategy parameter tuning. In: Proceedings of IEEE Intl Conference on High Performance and Smart Computing. pp. 74–84

Sun YN, Wang HD, Xue B, Jin YC, Yen GG, Zhang MJ (2020) Surrogate-assisted evolutionary deep learning using an end-to-end random forest-based performance predictor. IEEE Trans Evol Comput 24:350–364

Takahama T, Sakai S (2006) Constrained optimization by the ε constrained differential evolution with gradient-based mutation and feasible elites. In: Proceedings of IEEE Congress on Evolutionary Computation. pp. 1–8

Takahama T, Sakai S (2010) Constrained optimization by the ε constrained differential evolution with an archive and gradient-based mutation. In: Proceedings of IEEE Congress on Evolutionary Computation. pp. 1–9

Takahama T, Sakai S (2012a) Efficient constrained optimization by the epsilon constrained rank-based differential evolution. In: Proceedings of IEEE Congress on Evolutionary Computation. IEEE, pp. 62–69

Takahama T, Sakai S (2012b) Large scale optimization by differential evolution with landscape modality detection and a diversity archive. In: Proceedings of IEEE Congress on Evolutionary Computation. IEEE, pp. 2842–2849

Talbi EG (2019) A unified view of parallel multi-objective evolutionary algorithms. J Parallel Distrib Comput 133:349–358

Tan B, Ma H, Mei Y, Zhang M (2020) A cooperative coevolution genetic programming hyper-heuristic approach for on-line resource allocation in container-based clouds. IEEE Transactions on Cloud Computing, pp. 1–1

Tan KC, Feng L, Jiang M (2021) Evolutionary transfer optimization—a new frontier in evolutionary computation research. IEEE Comput Intell Mag 16:22–33

Tanabe R, Fukunaga A (2013) Success-history based parameter adaptation for differential evolution. In: Proceedings of IEEE Congress on Evolutionary Computation. IEEE, pp. 71–78

Tang DB, Dai M, Salido MA, Giret A (2016) Energy-efficient dynamic scheduling for a flexible flow shop using an improved particle swarm optimization. Comput Ind 81:82–95

Tenne Y, Goh C (2010) Computational intelligence in expensive optimization problems. Springer Science & Business Media, Berlin

Tessema B, Yen GG (2009) An adaptive penalty formulation for constrained evolutionary optimization. IEEE Trans Syst Man Cybern Part A-Syst Humans 39:565–578

Thiele L, Miettinen K, Korhonen PJ, Molina J (2009) A preference-based evolutionary algorithm for multi-objective optimization. Evol Comput 17:411–436

Thomsen R (2004) Multimodal optimization using crowding-based differential evolution. In: Proceedings of IEEE Congress on Evolutionary Computation. IEEE, pp. 1382–1389

Tian J, Sun CL, Jin YC, Tan Y, Zeng JC (2016) A self-adaptive similarity-based fitness approximation for evolutionary optimization. In: Proceedings of IEEE Symposium Series on Computational Intelligence

Tian Y, Cheng R, Zhang XY, Su YS, Jin YC (2019) A strengthened dominance relation considering convergence and diversity for evolutionary many-objective optimization. IEEE Trans Evol Comput 23:331–345

Tinos R, Yang SX (2007) A self-organizing random immigrants genetic algorithm for dynamic optimization problems. Genet Program Evol Mach 8:255–286

Unger NJ, Ombuki-Berman BM, Engelbrecht AP (2013) Cooperative particle swarm optimization in dynamic environments. In: Proceedings of IEEE Symposium on Swarm Intelligence. IEEE, pp. 172–179

Vafashoar R, Meybodi MR (2020) A multi-population differential evolution algorithm based on cellular learning automata and evolutionary context information for optimization in dynamic environments. Appl Soft Comput 88:19

van den Bergh F, Engelbrecht AP (2004) A cooperative approach to particle swarm optimization. IEEE Trans Evol Comput 8:225–239

Vidanalage BDG, Toulabi MS, Filizadeh S (2018) Multimodal design optimization of v-shaped magnet ipm synchronous machines. IEEE Trans Energy Convers 33:1547–1556

Voutchkov I, Keane AJ, Bhaskar A, Olsen TM (2005) Weld sequence optimization: The use of surrogate models for solving sequential combinatorial problems. Comput Meth Appl Mech Eng 194:3535–3551

Wang Y, Cai ZX (2012a) Combining multiobjective optimization with differential evolution to solve constrained optimization problems. IEEE Trans Evol Comput 16:117–134

Wang Y, Cai ZX (2012b) A dynamic hybrid framework for constrained evolutionary optimization. IEEE Trans Syst Man Cybern Part B-Cybern 42:203–217

Wang HD, Jin YC (2020) A random forest-assisted evolutionary algorithm for data-driven constrained multiobjective combinatorial optimization of trauma systems. IEEE Trans Cybern 50:536–549

Wang HD, Yao X (2014) Corner sort for pareto-based many-objective optimization. IEEE Trans Cybern 44:92–102

Wang Y, Cai Z, Zhou Y, Zeng W (2008) An adaptive tradeoff model for constrained evolutionary optimization. IEEE Trans Evol Comput 12:80–92

Wang H, Rahnamayan S, Wu ZJ (2013a) Parallel differential evolution with self-adapting control parameters and generalized opposition-based learning for solving high-dimensional optimization problems. J Parallel Distrib Comput 73:62–73

Wang R, Purshouse RC, Fleming PJ (2013b) Preference-inspired coevolutionary algorithms for many-objective optimization. IEEE Trans Evol Comput 17:474–494

Wang Y, Li HX, Yen GG, Song W (2015) MOMMOP: Multiobjective optimization for locating multiple optimal solutions of multimodal optimization problems. IEEE Trans Cybern 45:830–843

Wang HD, Jin YC, Jansen JO (2016a) Data-driven surrogate-assisted multiobjective evolutionary optimization of a trauma system. IEEE Trans Evol Comput 20:939–952

Wang J, Zhang W, Zhang J (2016b) Cooperative differential evolution with multiple populations for multiobjective optimization. IEEE Trans Cybern 46:2848–2861

Wang Y, Wang BC, Li HX, Yen GG (2016c) Incorporating objective function information into the feasibility rule for constrained evolutionary optimization. IEEE Trans Cybern 46:2938–2952

Wang ZJ, Zhan ZH, Du KJ, Yu ZW, Zhang J (2016d) Orthogonal learning particle swarm optimization with variable relocation for dynamic optimization. In: Proceedings of IEEE Congress on Evolutionary Computation. IEEE, pp. 594–600

Wang HD, Jin YC, Doherty J (2017) Committee-based active learning for surrogate-assisted particle swarm optimization of expensive problems. IEEE Trans Cybern 47:2664–2677

Wang HD, Jin YC, Doherty J (2018a) A generic test suite for evolutionary multifidelity optimization. IEEE Trans Evol Comput 22:836–850

Wang ZJ et al (2018b) Dual-strategy differential evolution with affinity propagation clustering for multimodal optimization problems. IEEE Trans Evol Comput 22:894–908

Wang BC, Li HX, Li JP, Wang Y (2019a) Composite differential evolution for constrained evolutionary optimization. IEEE Trans Syst Man Cybern Syst 49:1482–1495

Wang HD, Jin YC, Sun CL, Doherty J (2019b) Offline data-driven evolutionary optimization using selective surrogate ensembles. IEEE Trans Evol Comput 23:203–216

Wang Y, Li JP, Xue XH, Wang BC (2020a) Utilizing the correlation between constraints and objective function for constrained evolutionary optimization. IEEE Trans Evol Comput 24:29–43

Wang ZJ, Zhan ZH, Lin Y, Yu WJ, Wang H, Kwong S, Zhang J (2020b) Automatic niching differential evolution with contour prediction approach for multimodal optimization problems. IEEE Trans Evol Comput 24:114–128

Wang ZJ, Zhan ZH, Yu WJ, Lin Y, Zhang J, Gu TL, Zhang J (2020c) Dynamic group learning distributed particle swarm optimization for large-scale optimization and its application in cloud workflow scheduling. IEEE Trans Cybern 50:2715–2729

Wang ZJ, Zhan ZH, Kwong S, Jin H, Zhang J (2021) Adaptive granularity learning distributed particle swarm optimization for large-scale optimization. IEEE Trans Cybern 51:1175–1188

Weber M, Neri F, Tirronen V (2009) Distributed differential evolution with explorative–exploitative population families. Genet Program Evol Mach 10:343–371

Weber M, Neri F, Tirronen V (2011) Shuffle or update parallel differential evolution for large-scale optimization. Soft Comput 15:2089–2107

Wei FF, Chen WN, Yang Q, Deng J, Luo XN, Jin H, Zhang J (2021) A classifier-assisted level-based learning swarm optimizer for expensive optimization. IEEE Trans Evol Comput 25:219–233

Wessing S, Preuss M, Rudolph G (2013) Niching by multiobjectivization with neighbor information: Trade-offs and benefits. In: Proceedings of IEEE Congress on Evolutionary Computation. IEEE, pp. 103–110

Willmes L, Back T, Jin YC, Sendhoff B (2003) Comparing neural networks and kriging for fitness approximation in evolutionary optimization. In: Proceedings of IEEE Congress on Evolutionary Computation. pp. 663–670

Woldesenbet YG, Yen GG (2009) Dynamic evolutionary algorithm with variable relocation. IEEE Trans Evol Comput 13:500–513

Wolpert DH, Macready WG (1997) No free lunch theorems for optimization. IEEE Trans Evol Comput 1:67–82

Wu LJ, Zhan Z, Kwong S, Zhang J (2021a) Real traffic distance-aware logistics scheduling. In: Proc. IEEE Int Conf Syst, Man, and Cybern

Wu SH, Du KJ, Zhan ZH, Wang H, Zhang J (2021b) Historical information-based differential evolution for dynamic optimization problems. In: Proceedings of IEEE Congress on Evolutionary Computation

Wu SH, Zhan ZH, Zhang J (2021c) SAFE: Scale-adaptive fitness evaluation method for expensive optimization problems. IEEE Trans Evol Comput 25:478–491

Xiao C, Soetanto D, Muttaqi K, Zhang M (2020) A parallel evolutionary strategy for the large-scale dynamic optimal reactive power flow. In: 2020 IEEE International Conference on Power Electronics, Smart Grid and Renewable Energy. pp. 1–6

Xie WC, Yu W, Zou XF (2013) Diversity-maintained differential evolution embedded with gradient-based local search. Soft Comput 17:1511–1535

Xiong MH, Xiong W, Liu CX (2019) A hybrid many-objective evolutionary algorithm with region preference for decision makers. IEEE Access 7:117699–117715

Xu W, Xu J, He D, Tan KC (2019) An evolutionary constraint-handling technique for parametric optimization of a cancer immunotherapy model. IEEE Trans Emerg Topics Comput Intell 3:151–162

Yang SX, Li CH (2010) A clustering particle swarm optimizer for locating and tracking multiple optima in dynamic environments. IEEE Trans Evol Comput 14:959–974

Yang ZY, Tang K, Yao X (2008a) Large scale evolutionary optimization using cooperative coevolution. Inf Sci 178:2985–2999

Yang ZY, Tang K, Yao X (2008b) Multilevel cooperative coevolution for large scale optimization. In: Proceedings of IEEE Congress on Evolutionary Computation. IEEE, pp. 1663–1670

Yang SX, Li MQ, Liu XH, Zheng JH (2013) A grid-based evolutionary algorithm for many-objective optimization. IEEE Trans Evol Comput 17:721–736

Yang Q, Xie HY, Chen WN, Zhang J (2016) Multiple parents guided differential evolution for large scale optimization. In: Proceedings of IEEE Congress on Evolutionary Computation. IEEE, pp. 3549–3556

Yang M, Omidvar MN, Li CH, Li XD, Cai ZH, Kazimipour B, Yao X (2017a) Efficient resource allocation in cooperative co-evolution for large-scale global optimization. IEEE Trans Evol Comput 21:493–505

Yang Q, Chen W, Gu T, Zhang H, Deng JD, Li Y, Zhang J (2017b) Segment-based predominant learning swarm optimizer for large-scale optimization. IEEE Trans Cybern 47:2896–2910

Yang Q, Chen WN, Li Y, Chen CLP, Xu XM, Zhang J (2017c) Multimodal estimation of distribution algorithms. IEEE Trans Cybern 47:636–650

Yang Q, Chen WN, Yu ZT, Gu TL, Li Y, Zhang HX, Zhang J (2017d) Adaptive multimodal continuous ant colony optimization. IEEE Trans Evol Comput 21:191–205

Yang Q, Chen WN, Da Deng J, Li Y, Gu TL, Zhang J (2018) A level-based learning swarm optimizer for large-scale optimization. IEEE Trans Evol Comput 22:578–594

Yang Q, Chen W, Gu T, Zhang H, Yuan H, Kwong S, Zhang J (2020) A distributed swarm optimizer with adaptive communication for large-scale optimization. IEEE Trans Cybern 50:3393–3408

Yao J, Kharma N, Grogono P (2010) Bi-objective multipopulation genetic algorithm for multimodal function optimization. IEEE Trans Evol Comput 14:80–102

Yao G, Ding Y, Jin Y, Hao K (2017) Endocrine-based coevolutionary multi-swarm for multi-objective workflow scheduling in a cloud system. Soft Comput 21:4309–4322

Yeniay O (2005) Penalty function methods for constrained optimization with genetic algorithms. Math Comput Appl 10:45–56

Yildiz YE, Topal AO (2019) Large scale continuous global optimization based on micro differential evolution with local directional search. Inf Sci 477:533–544

Yin S, Kaynak O (2015) Big data for modern industry: challenges and trends. In: Proceedings of the IEEE, Feb. vol 2. pp. 143–146

Yin Y, Zhao YH, Li H, Dong XJ (2021) Multi-objective evolutionary clustering for large-scale dynamic community detection. Inf Sci 549:269–287

Yu G, Jin Y, Olhofer M (2021) A multiobjective evolutionary algorithm for finding knee regions using two localized dominance relationships. IEEE Trans Evol Comput 25:145–158

Yuan Y, Xu H, Wang B, Yao X (2016) A new dominance relation-based evolutionary algorithm for many-objective optimization. IEEE Trans Evol Comput 20:16–37

Yuan Y, Ong YS, Gupta A, Xu H (2018) Objective reduction in many-objective optimization: evolutionary multiobjective approaches and comprehensive analysis. IEEE Trans Evol Comput 22:189–210

Zaman F, Elsayed SM, Ray T, Sarker RA (2018) Evolutionary algorithms for finding nash equilibria in electricity markets. IEEE Trans Evol Comput 22:536–549

Zhan ZH et al. (2021) Matrix-based evolutionary computation. IEEE Transactions on Emerging Topics in Computational Intelligence, 1–14

Zhan ZH, Zhang J, Li Y, Chung HSH (2009) Adaptive particle swarm optimization. IEEE Trans Syst Man Cybern Part B-Cybern 39:1362–1381

Zhan ZH, Zhang J, Li Y, Shi YH (2011) Orthogonal learning particle swarm optimization. IEEE Trans Evol Comput 15:832–847

Zhan ZH, Li JJ, Cao JN, Zhang J, Chung HSH, Shi YH (2013) Multiple populations for multiple objectives: a coevolutionary technique for solving multiobjective optimization problems. IEEE Trans Cybern 43:445–463

Zhan ZH, Li JJ, Zhang J (2014) Adaptive particle swarm optimization with variable relocation for dynamic optimization problems. In: Proceedings of IEEE Congress on Evolutionary Computation. IEEE, pp 1565–1570.

Zhan ZH, Liu XF, Gong YJ, Zhang J, Chung HSH, Li Y (2015) Cloud computing resource scheduling and a survey of its evolutionary approaches. ACM Comput Surv 47:1–33

Zhan ZH et al (2017) Cloudde: a heterogeneous differential evolution algorithm and its distributed cloud version. IEEE Trans Parallel Distrib Syst 28:704–716

Zhan ZH, Wang ZJ, Jin H, Zhang J (2020) Adaptive distributed differential evolution. IEEE Trans Cybern 50:4633–4647

Zhang YF, Chiang HD (2017) A novel consensus-based particle swarm optimization-assisted trust-tech methodology for large-scale global optimization. IEEE Trans Cybern 47:2717–2729

Zhang JQ, Sanderson AC (2009) Jade: Adaptive differential evolution with optional external archive. IEEE Trans Evol Comput 13:945–958

Zhang J et al (2011) Evolutionary computation meets machine learning: a survey. IEEE Comput Intell Mag 6:68–75

Zhang WW, Yen GG, He ZS (2014) Constrained optimization via artificial immune system. IEEE Trans Cybern 44:185–198

Zhang X, Tian Y, Cheng R, Jin Y (2015a) An efficient approach to nondominated sorting for evolutionary multiobjective optimization. IEEE Trans Evol Comput 19:201–213

Zhang XY, Tian Y, Jin YC (2015b) A knee point-driven evolutionary algorithm for many-objective optimization. IEEE Trans Evol Comput 19:761–776

Zhang XY, Tian Y, Cheng R, Jin YC (2016) Empirical analysis of a tree-based efficient non-dominated sorting approach for many-objective optimization. In: Proceedings of IEEE Symposium Series on Computational Intelligence. pp. 1–8

Zhang L et al (2019a) Cooperative artificial bee colony algorithm with multiple populations for interval multiobjective optimization problems. IEEE Trans Fuzzy Syst 27:1052–1065

Zhang WW, Zhang WZ, Yen GG, Jing HL (2019b) A cluster-based clonal selection algorithm for optimization in dynamic environment. Swarm Evol Comput 50:100454

Zhang X, Gong Y, Lin Y, Zhang J, Kwong S, Zhang J (2019c) Dynamic cooperative coevolution for large scale optimization. IEEE Trans Evol Comput 23:935–948

Zhang K, Xu Z, Xie S, Yen GG (2020a) Evolution strategy-based many-objective evolutionary algorithm through vector equilibrium. IEEE Trans Cybern, 1–13

Zhang X, Du KJ, Zhan ZH, Kwong S, Gu TL, Zhang J (2020b) Cooperative coevolutionary bare-bones particle swarm optimization with function independent decomposition for large-scale supply chain network design with uncertainties. IEEE Trans Cybern 50:4454–4468

Zhang X, Zhan ZH, Zhang J (2020c) Adaptive population differential evolution with dual control strategy for large-scale global optimization problems. In: Proceedings of IEEE Congress on Evolutionary Computation. pp. 1–7

Zhang F, Mei Y, Nguyen S, Zhang M (2021) Collaborative multifidelity-based surrogate models for genetic programming in dynamic flexible job shop scheduling. IEEE Trans Cybern, 1–15

Zhao SZ, Liang JJ, Suganthan PN, Tasgetiren MF (2008) Dynamic multi-swarm particle swarm optimizer with local search for large scale global optimization. In: Proceedings of IEEE Congress on Evolutionary Computation. IEEE, pp. 3845–3852

Zhao SZ, Suganthan PN, Das S (2011) Self-adaptive differential evolution with multi-trajectory search for large-scale optimization. Soft Comput 15:2175–2185

Zhao H et al (2020) Local binary pattern-based adaptive differential evolution for multimodal optimization problems. IEEE Trans Cybern 50:3343–3357

Zhao H, Chen ZG, Zhan ZH, Kwong S, Zhang J (2021) Multiple populations co-evolutionary particle swarm optimization for multi-objective cardinality constrained portfolio optimization problem. Neurocomputing 430:58–70

Zheng XL, Wang L (2018) A collaborative multiobjective fruit fly optimization algorithm for the resource constrained unrelated parallel machine green scheduling problem. IEEE Trans Syst Man Cybern Syst 48:790–800

Zhou ZZ, Ong YS, Nguyen MH, Lim D (2005) A study on polynomial regression and gaussian process global surrogate model in hierarchical surrogate-assisted evolutionary algorithm. In: Proceedings of IEEE Congress on Evolutionary Computation. pp. 2832–2839

Zhou AM, Jin YC, Zhang QF (2014) A population prediction strategy for evolutionary dynamic multiobjective optimization. IEEE Trans Cybern 44:40–53

Zhou YZ, Yi WC, Gao L, Li XY (2017) Adaptive differential evolution with sorting crossover rate for continuous optimization problems. IEEE Trans Cybern 47:2742–2753

Zhou S, Zhan Z, Chen Z, Kwong S, Zhang J (2020) A multi-objective ant colony system algorithm for airline crew rostering problem with fairness and satisfaction. IEEE Trans Intell Transp Syst, 1–15

Zhu Z, Chen L, Yuan C, Xia C (2018) Global replacement-based differential evolution with neighbor-based memory for dynamic optimization. Appl Intell 48:3280–3294

Zou XF, Chen Y, Liu MZ, Kang LS (2008) A new evolutionary algorithm for solving many-objective optimization problems. IEEE Trans Syst Man Cybern Part B-Cybern 38:1402–1412

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

Zhi-Hui Zhan[1,2] · Lin Shi[1,2] · Kay Chen Tan[3] · Jun Zhang[4,5]

[1] School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China

[2] Pazhou Laboratory, Guangzhou 510330, China

[3] Department of Computing, The Hong Kong Polytechnic University, Hung Hom, Hong Kong

[4] Hanyang University, Ansan, South Korea

[5] Chaoyang University of Technology, Taichung City, Taiwan