Swarm and Evolutionary Computation 44 (2019) 695–711

Contents lists available at ScienceDirect

Swarm and Evolutionary Computation

journal homepage: www.elsevier.com/locate/swevo

Ensemble strategies for population-based optimization algorithms – A survey

Guohua Wu [a], Rammohan Mallipeddi [b,*], Ponnuthurai Nagaratnam Suganthan [c,**]

[a] School of Traffic and Transportation Engineering, Central South University, Changsha 410075, PR China
[b] School of Electronics Engineering, Kyungpook National University, Taegu, 702 701, South Korea
[c] School of Electrical and Electronic Engineering, Nanyang Technological University, 639798, Singapore

ARTICLE INFO

Keywords:
Ensemble of algorithms
No free lunch
Population-based optimization algorithms
Numerical optimization
Evolutionary algorithm
Swarm intelligence
Parameter/operator/strategy adaptation
Optimization algorithmic configuration adaptation
Hyper-heuristics
Island models
Adaptive operator selection
Multi-operator/multi-method approaches

ABSTRACT

In population-based optimization algorithms (POAs), given an optimization problem, the quality of the solutions depends heavily on the selection of algorithms, strategies and associated parameter combinations, constraint handling method, local search method, surrogate model, niching method, etc. In the literature, there exist several alternatives corresponding to each aspect of configuring a population-based algorithm such as one-point/two-points/uniform crossover operators, tournament/ranking/stochastic uniform sampling selection methods, Gaussian/Levy/Cauchy mutation operators, clearing/crowding/sharing based niching algorithms, adaptive penalty/epsilon/superiority of feasible constraint handling approaches, associated parameter values and so on. In POA literature, No Free Lunch (NFL) theorem has been well-documented and therefore, to effectively solve a given optimization problem, an appropriate configuration is necessary. But, the trial and error approach for the appropriate configuration may be impractical because at different stages of evolution, the most appropriate configurations could be different depending on the characteristics of the current search region for a given problem.

Recently, the concept of incorporating ensemble strategies into POAs has become popular so that the process of configuring an optimization algorithm can benefit from both the availability of diverse approaches at different stages and alleviate the computationally intensive offline tuning. In addition, algorithmic components of different advantages could support one another during the optimization process, such that the ensemble of them could potentially result in a versatile POA. This paper provides a survey on the use of ensemble strategies in POAs. In addition, we also provide an overview of similar methods in the literature such as hyper-heuristics, island models, adaptive operator selection, etc. and compare them with the ensemble strategies in the context of POAs.

1. Introduction

Optimization plays important roles in scientific research, management, and industry because numerous real-world problems can be essentially modeled as optimization tasks. “Traditional” mathematical programming methods (e.g., gradient-based methods) are no longer completely effective in solving complex optimization problems characterized by multi-modality, discontinuity, and noise. Different kinds of population-based optimization algorithms (POAs) have been emerging as promising alternatives in response to these challenges.

In POAs, multiple individuals search the solution space cooperatively and globally with operators and mechanisms like mutation, crossover, selection, information sharing and learning. In addition, randomness is usually embedded into one or more operators, so that POAs possess the capability to escape local optimal points and better explore the search space. Compared with other optimization algorithms, the most important characteristics of POAs could be three-fold. First, it searches the solution space through multiple points (solutions or individuals) simultaneously. Second, they have mechanisms for information sharing and interactive learning between individuals with diverse search behaviors. Third, POAs are stochastic as randomness is usually incorporated into the search behaviors including mutation, crossover, selection and others. The skeleton of a POA is depicted in Fig. 1.

POAs could be roughly categorized into evolutionary algorithms

* Corresponding author.
** Corresponding author.
E-mail addresses: guohuawu@nudt.edu.cn (G. Wu), mallipeddi.ram@gmail.com (R. Mallipeddi), epnsugan@ntu.edu.sg (P.N. Suganthan).

https://doi.org/10.1016/j.swevo.2018.08.015
Received 14 January 2017; Received in revised form 7 August 2018; Accepted 22 August 2018
Available online 5 September 2018

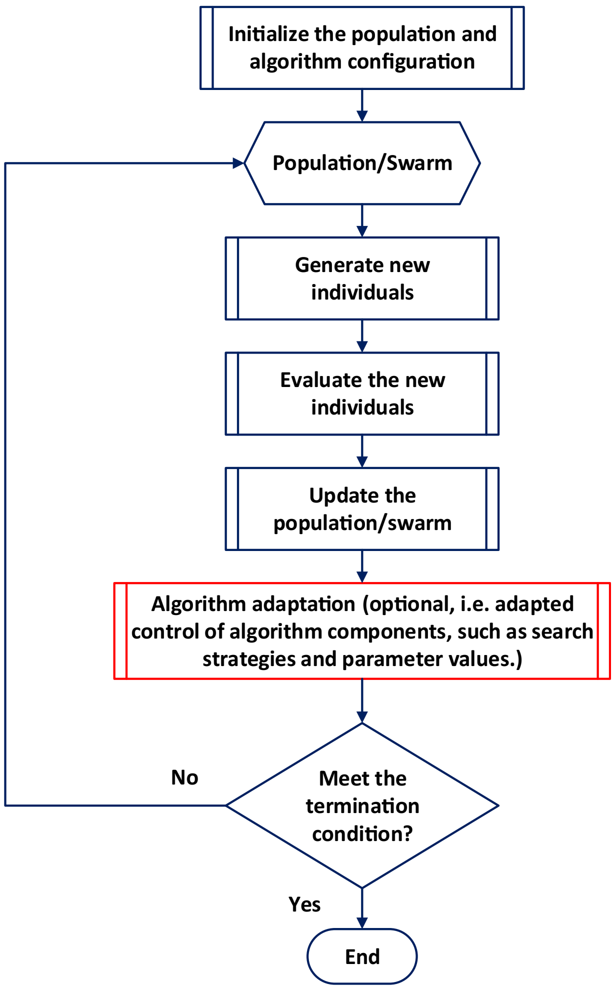

Fig. 1. A general framework of population based algorithms.

(EAs) and swarm intelligence algorithms (SIAs). Classical and popular EAs include Genetic Algorithm (GA) [1], Evolution Strategy (ES) [2,3], Evolution Programming (EP) [4,5] and Differential Evolution (DE) [6]. Besides, popular SIAs include Ant Colony Optimization (ACO) [7], Particle Swarm Optimization (PSO) [8], Artificial Colony Bee (ABC) [9], Artificial Immune Systems (AIS) [10], Across Neighborhood Search (ANS) [11] and several others. These POAs are well-known for their ability to solve optimization problems with multiple objectives and/or constraints that may not always be continuous and/or differentiable and may be characterized by chaotic disturbances, randomness and complex non-linear dynamics. POAs being stochastic "Generate-and-Test" algorithms only need the objective and constraint function values and do not require information regarding their characteristics, thus POAs are particularly suitable for Black-box optimization.

POAs have attracted much attention in recent decades and made continuous progress [12]. In addition to the great achievement POAs made in the IEEE CEC series of academic optimization competitions [13–17] and black-box optimization competitions (BBComp) (https://bbcomp.ini.rub.de/), POAs have been widely applied to various real-world applications [18], such as satellite scheduling [19], image classification [20], feature selection [21], planning of Earth-observation system [22], configuration of satellite orbit [23] and medical diagnosis [24].

Numerous methods and strategies have been proposed to enhance the precision, time efficiency and robustness of POAs. The ensemble strategy is one of the most promising approaches and has resulted in many efficient and versatile POA variants for unconstrained single objective numerical optimization, constrained optimization, multi-objective optimization, niching, etc. [25–27]. Motivations of using ensemble strategies in the design of POAs can be summarized as follows.

First, the No Free Lunch (NFL) Theorem [28] states that theoretically there exists no algorithm which is superior to the other algorithms in solving all possible optimization problems. In practice, NFL theorem indicate that it is impossible to design an algorithm (e.g. population-based optimization algorithm) that is more effective than all other algorithms in solving diverse optimization problems with different characteristics. Evidences frequently show that researchers now hardly can design a POA that can generate better results than all other state-of-the-art comparative POAs for a CEC or BBComp optimization benchmark set, which contains multiple types of optimization problems [26,29]. However, researchers are always devoted to developing versatile POAs that are suited to many kinds of optimization problems. A practicing engineer may not know which algorithm is good for a new optimization problem and may find it hard to tailor an existing algorithm to meet their special requirements. Furthermore, to select an efficient one from thousands of candidate algorithms is time-consuming. This sounds a frustrating contradiction. Is it possible there exists a practical and useful algorithm that can deal with a set of optimization problems of different characteristics? Our answer is yes. This is because although NFL tells that there is no algorithm being efficient for all possible optimization problems, in practice the problems considered are always a subset of the all. Hence, it is quite possible to develop a POA that is efficient for this specialized problem subset. How to design such a versatile POA is exactly a key concern of the algorithm researchers. Ensemble strategy provides an effective tool and paradigm to implement versatile POAs. In an ensemble POA, there are multiple search operators (e.g. mutation), parameters, constraint handling techniques or neighborhood structures, which generally have different characteristics and capabilities (e.g. exploration and exploitation) and therefore are suited to different types of optimization problems. As a result, a sophisticatedly designed ensemble POA with effective and distinguished ensemble components is potentially able to deal with different kinds of optimization problems. Note that, in this study we call these constituent search operators, parameters, constraint handling techniques or neighborhood structures in one ensemble as components of the ensemble.

Second, to improve the probability of finding the optimal solution for a hard optimization problem with complex landscapes, it is better to use diverse search (sampling) approaches. The same search approach may make the POA follow similar trajectories and be trapped in one of the local optima. Therefore, ensemble of multiple strategies could make POA particularly efficient for complicated optimization problems (e.g. composition functions of CEC 2014 single objective optimization benchmark) [26]. In addition, fixed strategies may not be most appropriate for the entire search process. Therefore, during the optimization process the search strategy needs to be updated to suit the search process as the search landscape changes during the population evolution towards the global optimal solution. Ensemble of multiple strategies with proper adaptation mechanism could enable a POA to have higher probability to select the most appropriate strategy during the optimization process [28]. Moreover, ensemble could also make search strategies of different capabilities support each other thus greatly strengthen the performance of a POA. For example, in an ensemble, exploration search strategies could find more unvisited promising areas which could be further refined with exploiting search strategies [29].

Third, given an optimization problem, the quality of solution to the problem and convergence speed of an algorithm to the optimal solution highly depends on the parameter and search strategy configuration of the optimization algorithm. Configuring the parameters and search strategies of a POA refers to finding the best combination of operators, parameter values and search strategies prior to or during the optimization process to maximize the performance of the algorithm on the given problem. Algorithmic configuration can be performed prior to or during the optimization process. When the algorithmic configuration is conducted prior to the optimization process, it is usually referred to as tuning the algorithm. On the other hand, when the algorithmic configuration is conducted during the optimization process, it is referred to as adapted operator, parameter and strategy control. It should be noted that traditional algorithmic configuration is to search for a fixed combination of strategies, operators and parameter values. However, traditional algorithmic configuration methods based on trial and error are usually time-consuming inefficient and incapable of changing during the evolution. Adapted control approaches dynamically adjust the operators, parameters and strategies of POAs according to the optimization states and have attracted much attention recently [32]. Ensemble of a set of promising candidate parameters values and strategies can properly realize the adapted control of parameters, operators and strategies, thereby alleviating the burden of configuring and selecting parameters and strategies for POAs.

The ensemble concept with respect to a POA can be defined as – a combination of different strategies, operators, parameter values and methods (with a single or parallel populations) referred to as an ensemble can provide better results on a set of optimization problems compared to a single set of strategies, operators, parameter values and methods. It should be noted that in machine learning, researchers use the concept of ensemble learning where multiple diverse models are combined to form an ensemble. For instance, neural network ensemble is a successful learning paradigm where a collection of a finite number of neural networks trained for the same task are combined to achieve a better performance on the same task, and has successful applications in diverse areas [30]. The basic idea of ensemble learning is based on the intuition that a group of 'unstable and diverse' learning algorithms can be combined to obtain an overall better learning algorithm. The extension of the ensemble concept to stochastic population-based optimization aims to construct a "stable" optimization algorithm by appropriate combination of "unstable and diverse" stochastic optimization algorithms [31].

It should be noted that there are some other concepts related to ensemble, including multi-strategy [25], multi-method [32], algorithm portfolios [33], hyper-heuristic [34], memetic algorithm [35] and hybrid algorithm [36]. In this survey, we take the concepts of multi-strategy and multi-method under the umbrella of ensemble. This is because multi-strategy can be viewed as a special case of the low-level ensemble of algorithmic components, while multi-method can be viewed as special cases of high-level ensemble of different EA variants. In contrast, the term of memetic algorithm (MA) was proposed by Moscato [37] as being a paradigm for integrating EAs with one or more refinement methods [35]. In MA, population based EAs (e.g DE, PSO and ABC) are taken as global search techniques while refinement methods (e.g. Nelder-Mead simplex search method, Hill Climber with Sidestep and trust-region derivative-free methods) play the role of local search [38]. In response to the challenge that the performance of an algorithm may vary significantly from problem to problem, algorithm portfolio attempts to find a less risky way to distribute the time among multiple different algorithms [33,39]. A hyper-heuristic is a methodology which, when given a particular problem instance or a class of instances, and a number of low-level heuristics (or their components), automatically produces an adequate combination of the provided components to effectively solve the given problem(s) [34]. Burke et al. categorized hyper-heuristic into heuristic selection and heuristic generation [40]. It can be perceived that hyper-heuristic aims to automatically develop new search algorithms based on a pool of low-level heuristics. Hybrid algorithm is a broader concept that may cover the other interrelated ones. Nevertheless, there is still a difference between ensemble algorithm and hybrid algorithm. For example, a differential evolution algorithm with ensemble of multiple mutation strategies is an ensemble algorithm while it would not be classified as hybrid algorithm. On the other hand, a particle swarm optimizer combined with a crossover operator is a traditional hybrid algorithm instead of an ensemble algorithm while simultaneous usage of multiple crossover strategies will make it a hybrid ensemble method.

Ensemble POAs have attracted much attention and resulted in encouraging achievements during the last decade. Different ensemble strategies have been proposed for low-level ensemble of algorithmic components including multiple search strategies, parameter values, etc. as well as high-level ensemble of multiple EA variants. Ensemble strategies have been incorporated into differential evolution (DE) [25,26,28,41,42], particle swarm optimization (PSO) [43–45], artificial bee colony (ABC) [46–48], biogeography-based optimization (BBO) [49–51] and so on. In addition, ensemble strategies are widely applied to different optimization areas, such as bound constrained single objective optimization [26,28,52], constrained optimization [53–56], multi-objective optimization [27,57,58], dynamic optimization [59,60], multi-modal optimization [61–63].

This paper reviews the state-of-the-art in ensemble POAs and provide readers a comprehensive picture of several aspects of ensemble POAs including taxonomy, implementation techniques, optimization areas, comparison with other interrelated terminologies and discussions on the possible future research. An overview of the concerned aspects of the ensemble strategy is given in Fig. 2.

The rest of the paper is structured as follows. Section 2 introduces the taxonomy according to the ensemble level. Section 3 surveys the implementation techniques of ensemble POA. Section 4 reviews the applications of ensemble strategies to different optimization areas. Section 5 discusses the relations and differences between the ensemble strategy and other related concepts. Section 6 provides some potential research directions of the ensemble strategy. Section 7 concludes the paper.

2. Ensemble of different levels

According to the characteristics of the constituent elements, ensembles in POAs can be roughly classified into low-level ensemble and high-level ensemble. The low-level ensemble is the ensemble of multiple algorithmic components, including search strategies, parameter values, constraint handling techniques, neighborhood structures, etc. In contrast, the high-level ensemble refers to as the ensemble of different POA variants. It is also possible to include high and low ensemble strategies in a POA.

2.1. Low-level ensemble

Search strategies largely determine the characteristics of POAs. In other words, different search strategies are suited to different optimization problems. As a result, to design a versatile POA, an ensemble of multiple search strategies with different advantages provides a promising direction. Many DE variants with multiple mutation strategies were proposed [25,26,41,52,64–78]. Ensemble of both crossover and mutation strategies was further studied in the design of DE variants [79,80].

In addition to DE, ensembles of multiple search strategies/operators were also used in cuckoo algorithm [81], GA [56], PSO [44,45,82–87]. The ensembles of multiple evolutionary operators were effectively used for multi-objective optimization [71,88,89].

The performance of major POAs is significantly affected by the parameter configurations. It is quite clear that the most appropriate parameter values vary when POAs are applied to different optimization problems. In addition, the best parameter values of a POA may vary at different stages of the optimization process. However, parameter tuning prior to the optimization process is usually computationally expensive and dynamic control of parameter values is nontrivial. As a result, parameter self-adaptation approaches are becoming popular [90]. Ensemble of parameter values is a promising method to realize the automatic parameter control [91–93]. The simultaneous ensemble of search strategies and parameters values was investigated in Refs. [28,42,63]. In addition, Choi and Ahn studied the ensemble of search strategies, parameter values and different local search techniques [94]. It should be noted that traditional parameter adaptation/self-adaptation strategy, such as the history and archiving strategy [95–97], usually make the parameter values be changed in a continuous real-valued interval.

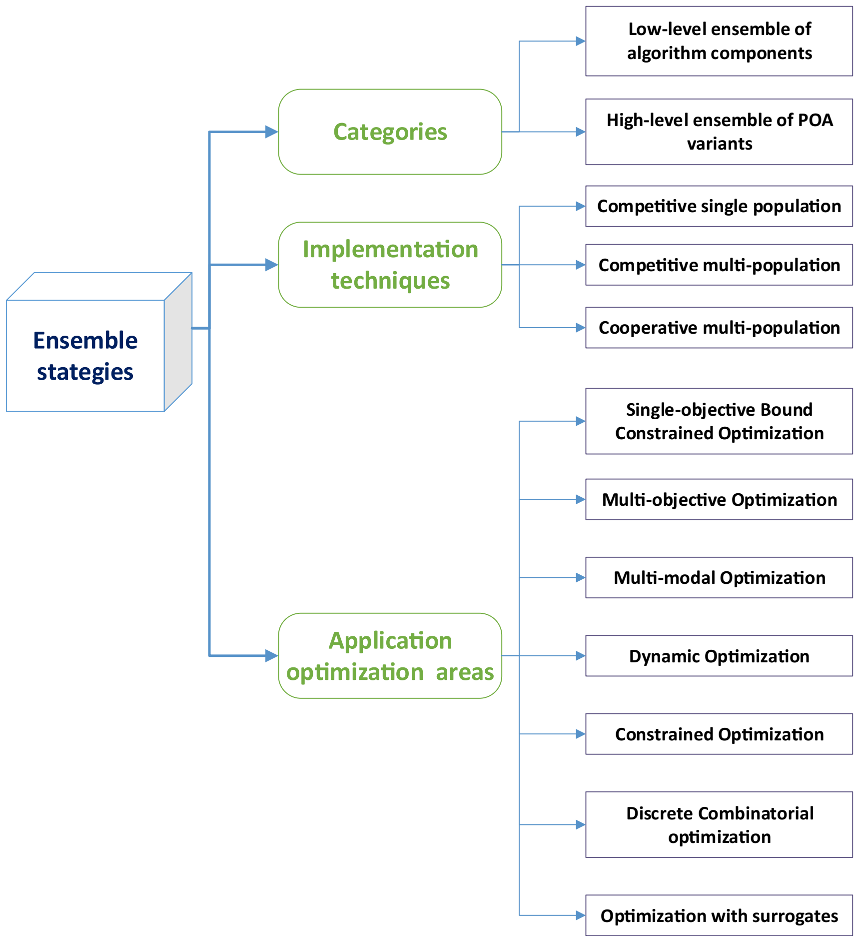

Fig. 2. An overview of different aspects of the ensemble strategy.

Parameters can be adapted by recording previously promising parameter values [41,95], dynamic adjustment according to evolutionary states [98], and being encoded in solutions and evolved with the optimization process [99]. The parameter adaptation can be at the population level [95,98] or individual level [38]. In contrast, the ensemble of parameter values usually requires the dynamic selection of an appropriate parameter configuration from a set of discretized parameter values [28]. The discretized candidate parameter values can be determined by previous studies of the research community [66]. Intuitively, compared to a single-valued parameter adaptation in a continuous real-valued interval, the ensemble of discrete parameter values enables the algorithm to switch rapidly to different ranges. Further, the selection discrete values can be varied by small randomization and enhanced by success rates based adaptation.

Ensembles of several other algorithmic components were also proposed by researchers, such as ensemble of surrogates [100,101], neighborhood sizes [27], and constraint handling techniques [53]. In addition, Karthikeyan and Baskar combined different immigrant strategies for GA [102]. Ma et al. presented the ensemble of migration models in biogeography-based optimization (BBO) [50]. Awad et al. combined different parameter values and parameter generation approaches to improve the performance of DE [103]. Elsayed et al. further proposed the simultaneous ensemble of search strategies and constraint handling techniques [54].

Many different approaches have been developed to realize the ensemble of components of POAs. For instance, the multi-population based mechanism [42,53,73,104–106] partitions the whole population into multiple subpopulations, each member (e.g. search strategy) in the ensemble owns a subpopulation and evolves independently. The information exchange may occur during the optimization process. Ensemble can be adapted through dynamic allocation of population resources [26,56], that is, the member in an ensemble performing better can be assigned with larger populations during the optimization process. The exploration and exploitation [29,72,107] mechanism means that in an ensemble, some members are responsible for exploration while others are used to exploit promising areas. In an ensemble, the usage frequencies of its members can be adapted with execution probabilities, which are determined by the former performance [25,41,70,74,77,82, 108], population diversity and fitness improvement [109]. The dynamic selection of the members in an ensemble can also be implemented in a sequential manner [110] or by some advanced techniques, such as reinforcement learning [83], artificial neural network regression [71], linear time-varying functions [111], game theory [112] and surrogate model [113]. Some studies determined the usage of members in an ensemble according to fitness of individuals [78,114], both fitness and positional information [115], and landscape characteristics [65].

Ensemble POAs generally show competitive performance. For instance, it is very impressive that the LSHADE-EpSin jointly won the CEC 2016 competition on single objective real-parameter numerical optimization based on the CEC 2014 benchmark set [103]. In LSHADE-EpSin, an ensemble of two kinds of parameter generation approaches was utilized.

2.2. High-level ensemble

It is known that different algorithms are suited to different types of optimization problems. To determine the most appropriate algorithm for a specific optimization problem or to make an algorithm effective for different kinds of optimization problems, the high-level ensemble of multiple methods provides an effective paradigm.

Several algorithms of ensembles of multiple DE variants have been proposed [116–119]. Elsayed et al. proposed the ensemble of multiple PSO variants with population resource reallocation for constrained optimization [55]. Lynn and Suganthan realized the ensemble of PSO variants for unconstrained single objective optimization [120]. A few works have further investigated the heterogeneous ensemble of different types of EAs [32,104,121]. Grobler et al. [122] compared different multi-method approaches including a population-based algorithm portfolio, a meta-hyper-heuristic and a bandit based operator selection method. They concluded that the meta-hyper heuristic was shown to outperform the other two algorithms. Elsayed et al. presented the ensemble of DE, PSO and ES variants, which are also ensembles of multiple operators [123–125].

The ensemble of multiple multi-objective evolutionary algorithms has been studied [126–129]. Vrugt and Robinson presented a multi-algorithm, genetically adaptive multi-objective (AMALGAM) to improve the multi-objective optimization process [130]. Ma et al. designed a multiple multi-objective BBO with ensemble of vector evaluated BBO, non-dominated sorting BBO, and niched Pareto BBO [51]. Mashwani and Salhi proposed a multi-objective optimization algorithm with the ensemble of pareto dominance-based concept and decomposition strategy based on dynamic resources allocation procedure [131].

It can be observed in literature that in a high-level of ensemble, different constituent algorithms could run alternatively [128], or through multi-population with migration and information exchanging mechanism [104], computational resource reallocation [131], execution probability adaptation [32,132], and constituent algorithm selection at time points [120,127]. In addition, Tang et al. proposed to use estimated performance matrix to evaluate and choose the constituent algorithm [39]. Akay et al. implemented the ensemble POAs by means of a message passing interface environment [133].

It is noticeable that UMOEAs-II that is an ensemble of multiple EAs [124] jointly won the CEC 2016 competition on Single Objective Real-Parameter Numerical Optimization based on the CEC 2014 benchmark set.

3. Techniques for realizing the ensemble

In POAs, the ensemble concept can be realized in a variety of ways and can be categorized depending on whether the constituent algorithms/parameters/strategies in the ensemble - a) cooperate or compete; and b) employs single or multiple populations. Generally, the cooperative ensembles are multi-population in nature, whereas competitive ensembles can be realized with single or multiple populations. Therefore, we can classify the ensemble approaches as follows:

a) Competitive single population ensembles
b) Competitive multi-population ensembles
c) Cooperative multi-population ensembles

3.1. Competitive single population ensembles

In a competitive single population ensemble, the individual methods/parameters/strategies compete for resources while operating on a single population. Depending on the way the resources are allocated to the constituent algorithms, during the evolution process, the competitive single population ensembles can be classified as [134]:

a) Local adaptation (Self-Adaptation): In local adaptation, each member in the population is assigned one of the constituent parameter/strategy, and then depending on its success, the population member is allowed to retain the same parameter/strategy or replace with another parameter/strategy from the pool. In other words, the parameter is encoded in the particular individual. Therefore, in a pool of parameters/strategies, the parameter/strategy that performs better gets attached to most of the population members depending on their location in the search space during the evolution process [134–136].
b) Global adaptation: In global adaptation, the constituent strategies/parameters in the ensemble are not assigned to the individuals but to the population as a whole [134]. In literature, the global adaptation has been realized in the following ways:
   1) The constituent strategies/parameters are assigned probabilities [27,32,63,103,137–140] or weights [141] (weighted ensemble of surrogates) which determine their applicability or amount of resources allocated to them. In addition, the probabilities or weights are dynamically updated during the evolution process after every few generations (can be updated every generation) depending on the performance of the individual strategies/parameters.
   2) In this approach, the strategies/parameters are stored in different archives referred to as initial, current and winning archives. At any given stage, when the current archive is empty, it is sampled from the initial and the winning archives. Each strategy/parameter in the current archive is tested and is moved to the winning archive if successful else it is deleted. The ratio of solutions taken from the initial and the wining archives is adjustable [142].

In competitive single population ensembles, while solving expensive optimization problems, surrogates or ensemble of surrogates [52,143] are employed to identify the best performing parameter/strategy from the pool.

3.2. Competitive multi-population ensembles

In competitive multi-population ensemble, each of the constituent methods/parameters is assigned a population. Competitive multi-population ensembles differ in the way the resources are allocated to the different populations.

1) Starting with equal population sizes, the size of the individual population is increased or decreased depending on the performance of the strategy/parameter/method assigned to the particular population [125,144,145].
2) Each constituent strategy/parameter in the population is assigned a larger and a smaller population referred to as indicator and reward populations, respectively. The resources allocated to the larger reward population depends on the performance of the respective indicator populations (in an allocated number of generations) [26].

3.3. Cooperative multi-population ensemble

Generally, the cooperative ensembles are multi-population in nature. In cooperative ensemble, the individual methods are allocated predefined (sometimes equal) resources and they try to cooperate with each other by exchanging information. The cooperative multi-population ensembles can be classified based on the way the different populations exchange information.

1) Uni-directional information flow: In Refs. [29,146], a multi-population with uni-directional information flow from an explorative population to an exploitative population was proposed. In other words, to preserve diversity and improve convergence, the exploitative population uses the information present in the explorative population while restricting the explorative population from using the information present in the exploitative population.
2) The individual populations can be arranged in a ring topology [147] or completely connected net topology [53,62,67,148–150] to determine the way in which they communicate the information. In addition to the frequency of information exchange, the amount and type of information that is being exchanged also vary.

a) In Refs. [53,62,148–150] each population has its associated parameters, strategies and selection rules. In addition, the offspring population produced by a population can go into the other populations if they are better according to the selection rules of that population. In other words, the populations interact by exchanging the produced offspring population and the information exchange is done during the selection phase. This method provides a better usage of each function evaluation.
b) In Ref. [147], communication is done by exchanging the elite member (fitness-based) and the communication is done based on a probability called migration probability.
c) In Ref. [67], the exchange of information is done at constant intervals and randomly selected individuals are passed on to the other populations.

3) In Ref. [93], after every few generations, the individual populations are combined and dynamically regrouped to facilitate the information exchange.

4. Ensemble methods in population-based optimization algorithms

In the last few decades, various population-based search algorithms such as differential evolution (DE), particle swarm optimization (PSO), evolution strategies (ES), and evolutionary programming (EP) were proposed each with its own strengths and weaknesses. In addition, corresponding to each algorithm there exist various combinations of strategies and parameter values. For example, to solve constrained problems, multi-modal problems, dynamic problems and multi-objective problems using population-based search algorithms it is essential to use constraint handling methods, niching methods, diversity enhancement techniques and non-domination sorting methods, where several alternatives corresponding to each of the techniques are available. In other words, to develop a population-based algorithm to solve a given problem, several alternatives are available at different stages such as selection of the algorithm, selection of strategies and parameters corresponding to the algorithm, etc. Hence, ensemble strategy being a general concept enabling appropriate selection for each problem, has been applied in various areas of optimization such as single objective unconstrained/constrained optimization, dynamic optimization, multimodal optimization, discrete optimization and multi-objective optimization and so on.

4.1. Ensemble methods in single-objective bound constrained optimization problems

The performance of population-based search algorithms on optimization problems with varying characteristics depends on the configuration of respective algorithm. With respect to single-objective bound constrained optimization problems the configuration refers to the appropriate selection of strategies and their associated control parameters. The process of selecting appropriate algorithmic configuration is not straight forward and requires extensive parameter tuning. In addition, during the evolution process different algorithmic configurations can be appropriate at different stages. Motivated by these observations, various works have been reported employing ensemble concepts to solve single-objective bound constrained optimization problems.

The performance of genetic algorithm (GA) depends on the appropriate selection of crossover strategy. To enhance the performance of GA, competitive single population ensemble strategies with local [134] and global [134]adaptations were explored. In GA with local adaptation [134], the crossover operator intended to be performed on the population member is encoded in the individual by using an extra bit. During crossover operation, actual operator employed depends on the last bits of the two selected individuals - if the last bits are same then the respective crossover operation is applied else one among them is randomly selected. It has been argued that encoding the constituents of the ensemble into the individual population members demonstrates improved performance, as different individuals are expected to follow different trajectories through the search space, requiring different operators for each trajectory. In GA with global adaptation [138], given different crossover operators, there exist different strategies for updating the operator application probabilities – 1) correlated to the occupancy rate of the solutions generated by the respective operator; 2) anti-correlated to the occupancy rate of the solutions generated by the respective operator, 3) maintaining occupancy rate of each crossover as close as possible, and 4) maintaining operator probabilities of all constituent method with an expected rate regardless of the occupancy rates. Through experimental analysis, it has been demonstrated that the first and the third strategies suit perfectly for travelling salesman problem and graph bisection problem, respectively.

Differential evolution (DE) is sensitive to the choice of the strategies (mutation and crossover) and the associated control parameter values (scale factor of mutation strategy and the crossover rate of crossover strategy) as they play an important role in balancing the exploration and exploitation capabilities of algorithm. In DE, ensemble strategies with varying characteristics have been explored – single population ensembles [28,52,103,137,143,151–154], multi population ensembles [26,29,67,155], ensembles with local adaptation [28,52,143,151,152], ensembles with global adaptation [103,137], cooperative ensembles [29,67,155] and competitive ensembles [26].

In DE, to balance the exploration and exploitation, the most common strategy is to employ a large population size with explorative strategies and parameters in the initial stages of the evolution; and to employ a smaller population with an exploitative strategies and parameters in the final stages. In DE algorithm with population reduction, different strategies can be employed depending on the population size [153,154]. In other words, if the population size is greater than a threshold then an explorative mutation strategy can be employed while an exploitative mutation strategy is employed if the population size is less than the threshold. In EAs, amount of exploration and exploitation to be done depends on the characteristics of the problem and therefore determining a threshold which allows us to switch between the exploration and exploitation may not be apt. Therefore, competitive single population ensemble DE with local adaptation (EPSDE) [28] with a pool of distinct mutation strategies (and crossover strategies [151]) along with a pool of values for each control parameter that coexist throughout the evolution process was proposed. In EPSDE, during the evolution, the individual strategies/control parameters compete with the other members in the pool to produce offspring. Later similar ideas have been explored with in the ABC framework [46,156,157]. Unlike in EPSDE where each population individual is assigned a combination of strategies and parameters picked from the respective pools, an ensemble DE where each target individual has an associated list of strategies and parameters was proposed [152]. During the evolution, a trial individual is generated by using strategies and parameters taken from the lists associated with the target vector and if the obtained trial individual is better than the target vector, the used strategies and parameter values are archived separately. The archived information is employed to update the pools associated with each target vector so that they can gradually self-adapt using the previous experiences to match different phases of evolution.

In DE, single population competitive ensemble with global adaptation was explored in Refs. [103,137], where the probability of selecting a combination of strategies and parameters to produce an offspring depends on the success rate of the combination. In addition, in DE the control parameters - namely the scale factor and the crossover rate are continuous parameters. Therefore, unlike in Refs. [28,151,152] where the control parameter pools are discrete, the control parameter values associated with each strategy are sampled form the mean and standard deviation values that are updated using the combinations that produce successful offspring [103,137].

In single population based DE ensembles, the exploration and exploitation are achieved by choosing an appropriate combination of strategies and parameters. In addition, the success of the combinations is determined by the improvement provided by the respective combinations. However, it is well-known that an exploitative combination is expected to provide frequent minor improvements while an explorative combination is expected to provide infrequent significant improvements. In other words, during the initial stages of the evolution depending on the type of improvements obtained the algorithm might get struck with exploitative strategies leading to degraded performance. Therefore, idea of using multiple populations to realize the ensemble was investigated [26,29,67,155]. The number of populations varies depending on the implementation. A multi population cooperative ensemble DE algorithm with two populations – one each for exploration and exploitation was proposed [29]. The exploitative population uses the information present in the explorative population and the vice versa is not permitted. In other words, the information flow is uni-directional from explorative to exploitation to make sure the explorative population is diverse even when exploitation group converges thus balancing the exploration and exploitation. When solving a large scale optimization problem, it is essential to maintain a balanced distribution of solutions with different potential that can guide the search process in different directions in future generations [67]. Therefore, a cooperative multi population ensemble with four populations that evolve individually with different strategies for a fixed number of generations (referred to as isolation phase) before exchanging the information (referred to as migration phase) was proposed [67]. In addition, a complete net topology structure was employed for migrating solutions between populations and the migrating solutions are selected in a uniform random manner. Unlike in Ref. [67], a cooperative multi population ensemble with frequent information exchange at the end of every generation was proposed in Ref. [155]. In this ensemble, at the end of every generation the individual populations are combined and then divided again into multiple populations depending on the Euclidean distance between the combined population members. In other words, during the process of dividing the combined population with size N into multiple populations with size N_s, a solution is randomly chosen and distance to the other members in the population is calculated in the normalized decision space. Then the nearest $(N_s - 1)$ solutions are selected to form the first group and are removed from the combined population. The process is repeated until the combined population is divided into (N/N_s) populations.

In DE, a competitive multi population ensemble with mutation strategies was proposed [26]. The algorithm contains three equally sized smaller indicator populations corresponding to each of the strategies employed and one much larger reward subpopulation. After evolving the three subpopulations for a certain number of generations, the best performing mutation strategy is identified based on the ratio between fitness improvements and consumed function evaluations. Then the reward subpopulation will be allocated to the best performing mutation strategy. The process of evolving the indicator populations and allocating the reward population to the best performing strategy repeats after a fixed number of generations. In addition, the control parameters corresponding to each mutation strategy are adapted independently.

In PSO, to balance the exploration and exploitation, a multi population cooperative ensemble with uni-directional information flow was proposed [146] was proposed where the population is divided into explorative and exploitative populations. In the explorative population, the exemplars are generated by only using the personal best experiences of the particles present in the explorative population. However, the exploitative population can make use of the personal best experiences of the explorative and exploitative populations to generate the exemplars. The uni-directional information flow helps in preserve the diversity in the explorative subpopulation even when the exploitative subpopulation converges prematurely. In addition, in PSO the search behavior depends on the learning strategy employed. Therefore, to adaptively adjust the search behavior of PSO, an adaptive ensemble of four different learning strategies was explored [45]. In the algorithm, each particle in the population acquires information by an ensemble of four strategies to determine the source of information – a) archived position of the global best position, b) its personal best position, c) the personal best position of a random particle, and d) a random position nearby. In the ensemble, each of the above strategies helps convergence, exploitation, exploration and escape local optima, respectively.

The performance of harmony search (HS) depends on the parameters such as pitch adjustment rate and maximum bandwidth. To improve the performance of HS, a single population competitive ensemble of parameter sets with diverse numerical values was proposed in Ref. [142], where the applicability of the parameters is self-adapted during the evolution process using global adaptation. To self-adapt the parameter, a set of three different memories referred to as initial parameter set list (IPSL), winning parameter set list (WPSL) and current parameter set list (CPSL) are maintained. IPSL contains a diverse set of parameter combinations. Initially, CPSL is sampled from IPSL. Then a combination is randomly picked from CPSL to check if it can improve the solutions. If the combination is successful it enters WPSL else is deleted. Once the CPSL is empty, it is filled with solutions taken from IPSL and WPSL. The ratio of solutions taken from IPSL and WPSL is tunable. For instance, in Ref. [23] 75% individuals of CPSL are from WPSL and 25% from IPSL. The authors claim that the EHS eliminates the trail-and-error search for the best single parameter set and enables the algorithm to benefit from the match between the parameter sets during the different search phases.

The performance of evolutionary programming (EP) depends on the employed mutation operators such as Gaussian, Lévy and Cauchy and the scale factor. For example, Gaussian mutation operator is better at searching in a local neighborhood while the Cauchy mutation performs better over a larger neighborhood. Motivated by these observations, an ensemble approach where each mutation operator has its associated population and every population benefits from every function call was proposed [149]. The multi population cooperative ensemble with information exchange in every generation enables us to benefit from different mutation operators with different parameter values whenever they are effective during different stages of the search process. Similar ideas have been explored in biogeography-based optimization (BBO) whose performance depends on – a) migration model, and b) mutation operator [49,158]. BBO with an ensemble of three migration models using three parallel populations was investigated in Ref. [158], while an ensemble with three migration models - a) nonlinear migration model based on sinusoidal curve, b) backup migration operator based on perturb operator and c) blended operator; and two mutation operators - differential mutation operator and Lévy local search was investigated in Ref. [49].

To solve the single objective optimization problems, few high-level ensembles have been investigated [32,125,139] in the literature. The high-level ensembles try to benefit from the multiple different search algorithms which have their own advantages while solving optimization problems with varying characteristics. A self-adaptive multi-method optimization algorithm comprising of covariance matrix adaptation evolution strategy, genetic algorithm, and particle swarm optimizer was proposed in Ref. [32], where the constituent algorithms run concurrently and benefit from each other through information exchange using a common population. The algorithm referred to as A Multi-algorithm Genetically Adaptive Method for Single Objective Optimization (AMALGAM-SO) implements a self-adaptive learning strategy to control the number of offspring that an individual algorithm is allowed to generate in each generation. In Ref. [139] a hybrid memetic algorithm based on multiple offspring framework was proposed to benefit from the capabilities of the individual algorithms. In this algorithm, the number of function evaluations allocated to each of the individual algorithms in a dynamical manner referred to as the high-level relay hybrid (HRH). A high-level multi population ensemble referred to as united multi-operator evolutionary algorithms (UMOEAs) was proposed in Refs. [125,145]. In UMOEAs, different populations evolve independently using different EAs with multi-operators. In this framework, the initial large population is divided into several populations of equal size and population uses one combination of search operators. During the evolution process, UMOEA emphasizes on the best performing multi-operator EA, as well as the search operator by adaptively varying the sizes of the populations. In other words, the size of population corresponding to the successful operators is increased, and at the same time the size of population corresponding to the unsuccessful operators is shrunk. The measure of success and failure of any combination of operators is decided based on changes in the fitness values, constraint violations, and the feasibility ratio of the subpopulations individuals. In addition, after a fixed number of generations, information is shared among the subpopulations, to encourage more effective searching.

4.2. Ensemble methods in multi-objective optimization

In population-based multi-objective optimization, for a group of solutions to adequately represent the whole Pareto Front it is essential to avoid the clustering of solutions so that the diversity can be improved. In multi-objective optimization literature, ε non-dominance sorting is one such technique that does not allow more than one solution with difference in all objective values less than ε to be non-dominated by each other. However, the effectiveness of ε non-domination sorting method depends on the selection of the individual ε values. In other words, different objectives and different optimization algorithms require different ε values to maintain the diversity of the population. Motivated by this observation, multi-objective particle swarm optimization algorithm (MOPSO) with an ensemble of ε parameter values and an ensemble of external archives was proposed [57] to alleviate the difficulty of tuning the numerical values of the ε parameters for every objective in diverse optimization problems.

Unlike Pareto dominance based multi-objective algorithms where the goal is to find a set of representative Pareto optimal solutions in a single run, multi-objective algorithms based on decomposition decomposes the problem into many single objective optimization sub problems and employs aggregation approaches. In MOEA/D, the objective of each sub problem is a weighted aggregation of all objectives in the multi-objective problem and the neighborhood relations between the sub problems are defined based on the distances among their aggregation weight vectors. Therefore, in MOEA/D, the neighborhood size (NS) plays a crucial role since each sub problem is optimized using the solutions associated with its neighboring sub problems. In literature, it has been demonstrated that different multi-objective problems need different NSs, and even for a particular problem, using different NSs at different search stages could improve the performance. In other words, a large NS is required for exploration (or to increase diversity) while a small NS will be apt for exploitation. In Ref. [27], an ensemble of different NSs with online self-adaptation was proposed (ENS-MOEA/D) to alleviate the need for expensive trial and error search for the appropriate NS parameter value. In addition, a MOEA/D with ensemble of crossover operators (simplex crossover operator and center of mass crossover operator) where the probability of employing each operator at any given time is updated in an adaptive manner was investigated [140] and implemented.

In multi-objective optimization, a high-level multi population ensemble comprising of vector evaluated biogeography-based optimization (VEBBO), non-dominated sorting biogeography-based optimization (NSBBO), and niched Pareto biogeography-based optimization (NPBBO) was proposed in Ref. [51]. The algorithm is implemented using parallel populations, where each population is assigned one of the four algorithms and generates its own offspring. All of the offspring populations are combined to select a fixed number of the best individuals, which updates all of the parent populations. In this way, ensemble algorithm always keeps the individuals that are generated by the more suitable constituent algorithm, leading to performance that is better than any constituent algorithm.

In multi-objective optimization, the performance of different multi-objective population-based algorithms on a given optimization problem is evaluated using some performance metrics. However, every metric is suitable to measure some problem-specific characteristics but cannot measure the performance comprehensively [58]. Therefore, for fair evaluation of population-based multi-objective algorithms a performance metric ensemble with double elimination tournament selection was proposed in Ref. [58]. The double elimination tournament selection operator compares the approximation fronts obtained by different population-based multi-objective algorithm in a statistically meaningful way. In addition, the elimination of a quality algorithm due to unfair comparison using a single metric can be avoided using ensemble of performance metrics. Hence, the ensemble of performance metrics was argued to provide a more comprehensive comparison among various population-based multi-objective algorithms compared to a single performance metric.

4.3. Ensemble methods in multi-modal optimization

In multi-modal optimization using population-based algorithms, niching methods are commonly employed to maintain a diverse population by forming subgroups. In literature, various niching methods such as crowding, sharing, and clearing have been proposed. However, given an optimization problem, it is often necessary to try out several niching methods and tune their parameters to find the best combination. In literature, it has been demonstrated that irrespective of the exhaustiveness of the parameter tuning, no single combination can be the best for problems with diverse properties. Therefore, unlike the general tendency where the whole population is subjected to the same niching algorithm, it would be beneficial to use different niching techniques simultaneously as the existing niching methods in the literature offer a wide choice.

An ensemble of niching algorithms (ENA) that uses several niching methods in parallel to preserve population diversity and to benefit from the best method was proposed [62]. This approach employs parallel populations for each niching method because each niching method has a distinct method to select the population of the next generation and each population can employ only one selection method.

In restricted tournament selection (Crowding DE is a special case with $w =$ population size), the setting of window size parameter ($\boldsymbol{w}$) is crucial and affects the performance of the algorithms. In other words, there is no unique value of $\boldsymbol{w}$ that can be suitable for problems with varying characteristics. Therefore, an ensemble of restricted tournament selection was proposed using parallel populations with different values for window size parameter [150].

Like the window size in restricted tournament selection, in multi-modal optimization using clearing, the appropriate setting of clearing radius is crucial. In Refs. [93,159], an ensemble approach where the population is divided into subpopulations and each subpopulation is assigned with a different clearing radius was proposed.

Like in most niching techniques, the main challenge in the speciation niching technique is to balance between exploration and exploitation. In Ref. [62], authors propose to enhance the exploration by employing arithmetic recombination with speciation and exploitation by employing neighborhood mutation with ensemble strategies. To form the ensemble, the authors have four different pools one for each of the strategies and parameters of DE such as mutation strategy, crossover strategy, scale factor and crossover rates. The application rate of each parameter value/strategy in the pools is self-adapted based on the performance of the individual parameter value/strategy as in Ref. [41].

4.4. Ensemble methods in dynamic optimization

In dynamic optimization, population-based algorithms should be able to detect environmental changes and then track the optimum. For population-based algorithms, dynamic optimization can be challenging because the population tends to converge and loses its ability to explore when the environment is static for a considerable amount of time. For the algorithm to be flexible enough to react to environmental changes, the algorithm needs to explore several potentially good regions in the search space while simultaneously performing the fine search around the best solutions found so far. Therefore, in dynamic optimization, it is essential to increase the diversity of the population and ensure that the algorithm does not converge so that it can react and evolve further when the environmental change is detected. To maintain population diversity when solving dynamic optimization problems, different diversity enhancement/preserving mechanisms have been proposed. However, no single diversity preserving/enhancing method is superior to other.

In dynamic optimization, the idea to employ an ensemble of memory-based archives from which the solutions can be introduced into the population to improve the diversity was proposed [59]. In other words, population diversity is enhanced by an ensemble of external archives that serve as short-term and long-term memories. In addition, the solutions in the archives can also be employed to effectively detect the environmental changes.

Solving a dynamic optimization problem becomes further complicated if it involves multiple objectives. In Ref. [60], a multi-objective population-based algorithm with an ensemble of different strategies was proposed. In this algorithm, an ensemble of three different strategies was employed to effectively address the following issues - (1) to achieve fast convergence after an environmental change; (2) to improve global search ability and (3) to improve diversity.

4.5. Ensemble methods in constrained optimization

In population-based constrained optimization, to effectively solve a problem by exploiting the information present in infeasible individuals it essential to select an appropriate constraint handling technique. However, according to the No Free Lunch theorem, it is difficult to pick a single constraint handling technique that can effectively solve diverse constrained optimization problems. In addition, while solving a single constrained problem different constraint handling methods can be effective during different stages of the search process. Motivated by these observations, an ensemble of different constraint handling techniques (ECHT) was proposed [53], where each constraint handling method has its own population and the offspring produced by one population not only compete with the corresponding population members but also with the members of the other populations in order to make effective use of the function evaluations. A similar idea was explored to solve constrained multi-objective optimization problems in Ref. [160].

To solve constrained optimization problems, an overall ensemble in DE framework (SAS-DE) [54] comprising of a) four mutation operators, b) two crossover operators and c) two constraint handling techniques was proposed. In other words, sixteen different combinations of strategies were employed. In SAS-DE, initially, each individual is randomly assigned with one of the sixteen combinations. During the evolution, better performing combinations are emphasized by ranking the combinations. In the population, for the first few individuals, the respective combinations are selected through tournament selection while the combinations are assigned randomly to the remaining population members. The combinations in the ensemble are ranked through an improvement index which considers both the objective function values and the constraint violations.

To overcome the inherent drawbacks of PSO in solving constraint problems, a self-adaptive high-level ensemble comprising different PSO variants was proposed in Ref. [55]. Initially, each PSO variant is assigned an equal number of individuals from the current population. However, in the subsequent generations, the number of solutions from the current population allocated to the PSO variants depends on their performance. In other words, to emphasize better-performing variants the number solutions assigned to each variant are changed based on a success measure. The success measure considers the fitness values, constraint violations and feasibility ratios of individuals in the sub-populations.

A general optimization framework that could be used for training and testing a self-adaptive multi-operator Genetic Algorithm (SAMO-GA) was proposed in Ref. [56]. To facilitate self-adaptation, an improvement index based on the solution quality, constraint violation and feasibility rate is considered. SAMO-GA starts with a random initial population which is further divided into four subpopulations of equal size. Each subpopulation evolves using a combination of assigned crossover and mutation operators; and the generated offspring are evaluated. Depending on the improvement indexes of each subpopulation their sizes are increased, decreased or kept unchanged. In addition, after every few generations (window size) the information is exchanged between the subpopulations by passing the best solutions. To design a generalized framework, a set of 60 benchmark problems with diverse characteristics were considered; and the pool of test problems were divided into three groups, and three iterations were performed. In each iteration, two groups were used for training and the third one was used for testing. From the experiments, the best combination was found to be - population size (80), minimum sub-population size (10% of population size), and window size (50 generations).

4.6. Ensemble methods for discrete combinatorial problems

The performance of a population-based optimization algorithm in solving optimization problems with discrete variables such as generalized travelling salesman problem (GTSP) or scheduling problems depends on – a) destruction and construction (DC) procedures that work as a mutation operator and b) crossover operator. To improve the performance of DE algorithm on discrete optimization problems by making effective use of different parameter values and crossover operators simultaneously, an ensemble of discrete DE algorithms where each parameter set and crossover operator assigned to one of the parallel populations [148] was proposed. In this algorithm, each parallel parent population does not only compete with offspring population generated by its own population but also the offspring populations generated by all other parallel populations which use different parameter settings and crossover operators.

In scheduling problems related to manufacturing, critical paths play an important role [161]. In a population, each solution can contain one or more critical paths and each operation on a critical path is called a critical operation. Optimization algorithms that are based on critical path theory, sort the machines available at the manufacturing unit based on the number of critical operations processed on them. In the ensemble based local search algorithm, first the ensemble local search is executed on the machine with the maximum critical operations. In Ref. [136], five local search operators - one insert, one swap, two inserts, two swaps, one insert and one swap, are stored in an operator pool. In the initial stages, each operator is assigned equal probability of being selected for generating a neighbor solution. The operators improving the current solutions are maintained in operator pool while the operators that fail to improve are deleted. Finally, the ensemble local search will be terminated when all operators have been deleted from the operator pool. Then the same process is repeated on the next machine. In Ref. [162], flexible job shop scheduling problem with new job insertion was solved using an ensemble of four heuristics.

4.7. Ensemble of surrogates

To optimize computationally expensive optimization problems [30], surrogate or meta-models or fitness landscape approximations have been employed in conjunction with population-based algorithms. In literature, it has been well-documented that different surrogate models perform well in different conditions [163] such as complexity of the problem and number of samples available to construct the model, etc. In surrogate-model based algorithms, the general tendency is to fit a high quality single surrogate model that can represent the data points. However, in general, before-hand it is unclear which surrogate model type suits the best a given problem, and in practice it is not feasible to try different algorithms due to the computational cost of doing function evaluations. In ensemble evolutionary algorithms, the surrogate models are generally used to evaluate the competitiveness of different constituent parameter/strategy combinations [143,52]. Therefore, the number of original function evaluations required can be reduced when solving expensive optimization problems.

Surrogate models are an approximation of the original functions and the accuracy or performance of the surrogate models depends on various factors such as availability of sufficient number of samples. However, it has been argued in Ref. [164], that the uncertainty introduced by approximation errors in the surrogate model can benefit as well as obstruct effective optimization search. In other words, since there is often no prior knowledge about the most appropriate surrogate for the problem of interest, the use of multiple meta-models appears to be an intuitive alternative. Therefore, instead of a single best surrogate model, the idea of using an ensemble of surrogate models was investigated in Ref. [141]. In the ensemble, each individual surrogate model is assigned a weight which is determined based on the errors of the individual surrogates. The main motivation behind employing a weighted ensemble of surrogate models is to emphasize characteristics of "good" models and to restrict the influence of "bad" models in the combination.

The different ensemble concepts discussed above are summarized in the following table.

5. Techniques related to ensemble approaches

To effectively solve a diverse set of problems with minimal tuning and make the algorithms less dependent on the parameters, different ideas have been proposed in the literature namely hyper-heuristics, Island models and adaptive operator selection (AOS). The different ideas including ensemble can be classified based on characteristics such as:

a) Number of populations/subpopulations employed
b) Possibility of parallel evolution
c) Usage of explicit memory
d) Facilitation and Frequency of information exchange

5.1. Hyper-heuristics

In contrast to traditional population-based optimization algorithms which operate directly on the solution space, hyper-heuristics search a space of heuristics. In other words, hyper-heuristics automate the heuristic design process based on the structure of the problem to be solved. Due to the process of automation, hyper-heuristics offer advantages such as – a) scope of discovering novel heuristics for a given problem that are unlikely to be invented by human analyst, b) different heuristics can create different subset of instances so that the results obtained on each are more likely to be better than those obtained by one general heuristic. Given an optimization problem, hyper-heuristics are capable of providing a "good enough", "soon enough", "cheap enough" [165] solution compared to the conventional population-based algorithms. Therefore, hyper-heuristics tend to generalize the development of optimization algorithms that can effectively handle a wide range of problem instances such as timetabling [166] without expensive manual parameter tuning.

In the literature, considering the nature of the heuristic search space, hyper-heuristics are classified into two types: a) hyper-heuristics to choose heuristics – choose the best heuristics given a set of low-level heuristics or produce the best combination from preexisting heuristics, and b) Hyper-Heuristics to Create Heuristics – generate low-level heuristics from a set of building blocks given by the user or heuristic generation methods that generate new heuristics using basic components of preexisting heuristics.

Hyper-heuristics to choose heuristics: The hyper-heuristics can be developed for different purposes such as identifying the best global search algorithm from a predefined set (simulated annealing, genetic algorithm, particle swarm optimization, and ant colony optimization) [167] and/or identifying the best local search algorithm from a predefined set (APRILS) [168]. In hyper-heuristics, to dynamically determine which low-level heuristic is to be used in finding better candidate solutions, the quality of the individual heuristics are evaluated through diversity and improvement detection operators [167]. In case-base hyper-heuristics, heuristics that worked well during the earlier situations or cases are memorized in a case-base. Then given a problem, depending on the similarity of the current problem with the ones in the case-base, the heuristic corresponding to the most similar case in the case-base is retrieved and refined to solve problem [169]. In Ref. [169], the similarity measure employed is a nearest neighbor approach where weighted sum of differences between features of a given problem and cases in the case-base is considered.

Some significant applications of this class of hyper-heuristics include scheduling algorithm for cloud computing systems [167], discovering dispatching rules for dynamic job shop scheduling [168], timetabling [169–171] and shipper rationalization problem [172].

Hyper-Heuristics to Create Heuristics: In hyper-heuristic literature, heuristics developed from building blocks or low-level operators were reported such as

a) A hyper-heuristic framework that integrates a set of low-level operators such as mutation operator of DE, Cauchy mutation of EP and CLPSO operators [173]. Among the set, the best operator is identified in a stochastic manner by the performance feedback during the process of problem solving, thus yielding a high overall performance on different instances. In other words, the evolutionary operators in the set compete with each other for the computational resources.
b) An automatic generator of novel local search heuristics for challenging satisfiability testing (SAT) problems from a set of local search operators [174]. In Ref. [174], the hyper-heuristic referred to as "CLASS" was shown to perform better than directly evolving a solution through evolutionary algorithms for SAT problems.
c) An adaptive way to maintain low-level parameters (LLPs) with a search-based algorithm [175]. The adaptive maintenance of LLPs alleviates the need for fine-tuning and also enhances the modularity and generality of hyper-heuristics.

Furthermore, this particular class of hyper-heuristics has been applied to evolve dispatching rules for the job shop problem [176], solve nurse rostering problem [177], evolve reusable heuristics for 2-D strip packing problem [165], generate a good quality heuristics for each instance of one-, two-, or three-dimensional knapsack and bin packing problems [178], solve different optimization problem instances related to emergency railway transportation planning [173], solve satisfiability testing (SAT) problems [174], grammatical evolution based hyper-heuristic for two well-known combinatorial optimization problems which are exam timetabling and capacitated vehicle routing problem [179] and a hyper-heuristic approach based on genetic programming for evolving an enhanced version of the Sloan algorithm [180].

5.2. Island models

Island models are also known as coarse-grained models or multi-deme models or distributed algorithms or portfolio optimization [181]. Island models parallelize (to reduce the computational times when solving hard optimization problems) the evolution process by splitting the population into multiple sub-populations called islands. The sub-populations in each island evolve independently for most of the time. However, they exchange solutions between the islands periodically through a process called migration. To enable communication between them, the islands are connected by a graph structure. In Island models, the following issues have been investigated in the literature.

1. Effect of different EAs running independently on each island: In Ref. [182], the authors analyzed the effect of migrating individuals

between the islands employing heterogeneous genetic operators and concluded that the heterogeneity enhances the inter-island evolution.
2. Adaptively allocating function evaluations to solve within a given budget: Motivated by the computational portfolios in Ref. [181], a population-based algorithm portfolio (PAP) was proposed [33] where the whole population is divided into several subpopulations that evolve in parallel employing different search algorithms and interact through migration. In addition, during the evolution process, the computation time (measured in terms of number of fitness evaluations) is divided between the constituent algorithms depending on their performance. In other words, PAP is a combination of different population-based search algorithms that interact, benefit and compete.
3. Effect of different representations in each island: In Ref. [183], authors proposed an island model that uses a different representation in each island (using standard binary encoding in one island and standard reflective Gray code in the other) and during the process of migration, the model transforms individuals from one representation to another. The authors claim that the heterogeneous representations in the islands help the population-based algorithm to escape from local optima and to solve problems that are difficult for single representation EAs.
4. Reducing computational time by parallelization and effect of different topologies: The effect of different topologies (such as ring, von Neumann grid and hypercube employed to interconnect the islands) on the computational speedup gained due to parallelization [184] and the solution quality was analyzed in Refs. [184,185]. The analysis demonstrated that even sparse topologies (for instance ring topology) provide a significant speedup for many functions [184]. In addition, guidelines on the number of processors that can result in the best guaranteed speedups and on the parameterization of population-based algorithms were presented. It was also observed that the robustness of the algorithm improves with the density of the topology [184]. In Ref. [186], an investigation on how the island models deal with plateaus and effect of frequency of migration on the speedup were reported.
5. Effect of migration policies: The effect of different migration policies on the diversity of the population in an island was investigated in Ref. [187]. In addition, the authors propose a migration policy based on the principal of multiculturality and claimed it to be better than traditional migration policies such as sending the best or a random individual. They also report that the improvement in the performance increases with the increase in the number of nodes involved and the difficulty of the problem.

5.3. Adaptive operator selection (AOS)

As mentioned in the Introduction, the performance of a population-based optimization algorithm depends critically on the selection and adaptation of the parameters during the evolution process. Adaptive operator selection (AOS) is a mechanism employed to determine the application rates of different operators in an online manner based on their recent performances. The success of an AOS framework depends on two main components: a) credit assignment - defines the reward that will be used to evaluate the quality of an operator after its application; and b) operator selection mechanism - selects one operator based on some rewards gained by the operators based on their performance. During the credit assignment as well as operator selection, it is essential to assign a higher probability to operators that produce better offspring (exploitation) and at the same time needs to maintain operators that perform poorly for the future search (exploration) by assigning minimum probability [188]. In population-based algorithm literature, this is referred to as exploration versus exploitation dilemma and to overcome it, different credit assignment and operator selection policies have been proposed.

Credit Assignment is an important ingredient in AOS and different credit assignment policies developed in the literature can be classified as follows:

- Credit assignment based on average fitness improvement of the offspring – most commonly employed [189].
- Extreme value-based credit assignment [189,190] - rewards each operator based on the extreme value of the fitness improvement obtained using the operator during the current or last few application instances (sliding window). Extreme value-based credit assignment works on the notion that rare but high fitness improvements are more important than frequent and moderate improvements.
- Credit assignment that takes into account the diversity of the population in addition to fitness improvements to achieve a compromise between exploitation and exploration [191].
- Rank-based credit assignment schemes such as area-under-curve (AUC) and sum-of-ranks (SR) [192].

In the literature, the operator selection is generally based on techniques such as

- Probability Matching [193] aims to match the operator probabilities to their relative reward from the environment.
- Adaptive Pursuit [193] aims at a greater increase in probability to the most effective operator, and corresponding decreased probability to the other operators. In other words, it aims to provide faster response to changes in the reward profile.
- Multi-Armed Bandit (MAB) [194] is inspired by game theory and selects operators according to some deterministic rules. In game theory, MAB tries to learn the best action while optimizing its cumulative reward during learning. However, in the context of parameter control it tries to find the best operators during the evolution while maximizing the fitness improvement during the evolution.
- Dynamic MAB algorithm [195] addresses the dynamic and unpredictable aspects of the AOS framework. Dynamic MAB reevaluates the bandit statistics from scratch whenever a change in the reward distribution is detected.
- Sliding MAB method [196] employs a sliding window to achieve a faster update of the quality measure of each operator to better fit the current stage of the search process.
- Hybridization of MAB algorithm with the statistical Page-Hinkley test can be used to detect the changes [195].

In Ref. [25], the authors analyze the performance of operator selection strategies such as probability matching and adaptive pursuit in conjunction with autonomous strategy selection in DE. In their study, the effects of four different credit assignment methods are also analyzed. A systematic comparison between different AOS schemes such as MAB, DMAB and Adaptive Pursuit is presented in Ref. [196].

Adaptive operator selection methods based on probabilities such as self-adaptive DE (SaDE) [41] and DE with dynamic parameter selection (DE-DPS) [25,42] were reported in the literature. In SaDE, each of the strategies/operators in the pool is initially assigned with identical application probabilities and during the evolution the application probability corresponding to each strategy/operator is updated depending on the number of successful offspring produced. In DE-DPS, each of the parameters related to the DE algorithm – amplification factor, crossover rate and population size is assigned a pool of values. Each individual in the population is assigned a combination by randomly picking the parameter values from the individual pools and the success rate of each combination is noted. As the evolution progresses, based on the success rate of the combinations, some of the combinations with weak performance are removed to enable the better performing combinations to be more frequently used.

Most of the probability based AOS algorithms [41,95,197] have been proposed in conjunction with single-objective optimization problems, in which the fitness improvement of an offspring solution against its parent can easily be measured. However, in Pareto-dominance-based multi-objective optimization, some solutions are incomparable and therefore it is not straightforward to define and compute the quality difference between two solutions. Hence, the application of AOS in multi-objective population-based algorithms is still relatively scarce with works such as JADE2 [198], MOSaDE [199], OW-MOSaDE [200], Adap-MODE [201], and MODE/SN [202].

In multi-objective population-based algorithm based on decomposition (MOEA/D), a multi-objective optimization problem is decomposed into many scalar optimization subproblems and optimized simultaneously. Thus, it is natural and feasible to use AOS in MOEA/D unlike Pareto-dominance multi-objective optimization algorithms. In Ref. [188], the authors proposed a bandit-based AOS method referred to as fitness-rate-rank-based multi-armed bandit (FRRMAB). FRRMAB uses a sliding window to record the recent fitness improvement rates achieved by the operators, while employing a decaying mechanism to increase the selection probability of the recent best operators.

5.4. Hybrid – AOS with hyper-heuristic

To address the issue of appropriate selection and updating of the parameters to suit the problem characteristics, a hybrid approach that combines hyper-heuristics and AOS mechanisms was proposed in Ref. [203]. In other words, in the proposed hybrid algorithm referred to as Adaptive Hyper-Heuristic, hyper-heuristic is employed to select the appropriate metaheuristic from a set of heuristics (Genetic Algorithm and Differential Evolution) and AOS is employed to dynamical update/select the appropriate operators corresponding to the metaheuristic selected (five operators of GA and four operators of DE). Two AOS schemes are considered: the Adaptive Pursuit and the Fitness Area Under Curve Multi-Armed Bandit.

5.5. Comparison between ensemble method and related approaches

Even though there are several methods in the literature to configure optimization algorithms that can effectively solve problems with diverse characteristics with minimal tuning, they have some similarities and differences. Table 1 summaries the differences and similarities between the ensemble-based algorithms and other related approaches (see Table 2).

Hyper-heuristics and adaptive operator selection approaches use single population where a group of heuristics/methods/operators compete for resources. In both the cases, the allocation of resources to the constituent heuristics/methods/operators is adaptive depending on their performance during previous generations. Because they employ single population, the concepts of information exchange and evolving solutions in parallel are not possible. Both these approaches employ explicit memory to store the constituent heuristics/methods/parameters. However, the explicit memory employed by hyper-heuristics is high, for instance in case-base hyper-heuristics.

The concept of Island models is based on multiple populations and parallelizes the evolution process. In addition, the information exchange between the multiple populations is done occasionally through a process called migration. In Island models, each population can employ its own POA, operators, etc. that can be adaptive too. In Island models with adaptive POAs, moderate explicit memory is needed to store the successful combination of methods/parameters.

However, ensemble-based approaches can use single population (EPSDE [28]) or multiple populations ([53,64,149]). Depending on the implementation (single or multiple populations), ensemble-based approaches can be suitable to implement parallel evolution. However, unlike in Island models where each population is connected by a graph structure to facilitate information exchange, ensemble-based approaches employ no specific rigid structure. In addition, communication between different populations is occasional in Island models to preserve diversity while the communication in ensemble-based approaches may be occasional (after every few generations) or frequent (every generation). In addition, similar to the other approaches, the ensemble-based approaches can be adaptive (EPSDE [151]) or non-adaptive (ECHT [53]). Depending on the level of adaptation being implemented, ensemble approaches may employ moderate explicit memory.

From the above discussion, it is clear that unlike other methods, ensemble enables the user to have many variations. For instance, an ensemble with parallel populations offers the flexibility of having different parent-offspring selection rules to obtain the next generation individuals while an ensemble with single population enables a single parent-offspring selection rule. In addition, unlike hyper-heuristics and island models, ensemble approaches do not divide the total number of function evaluations between the constituent algorithms/operators. Hence, each function evaluation in the ensemble configuration contributes to the evolution of the population whereas dividing function evaluations between the constituent algorithms/operators prevent the benefit of each function evaluation being enjoyed by the total population.

In general, ensemble-based approach allows variations such as single or multiple populations, adaptive or non-adaptive, frequent communication for information exchange without loss of diversity and can operate efficiently at both operator-/parameter-level (like adaptive operator selection) and algorithm-level (like hyper-heuristics).

Table 1
Ensemble Algorithms proposed and their characteristics.

Algorithm [Reference]	Characteristics
Single-objective bound constrained optimization	
DE [153,154]	Low-level, single population ensemble
GA [134], DE [28,151,152], ABC [46,156,157]	Low-level, competitive single population ensemble with local adaptation
GA [138], DE [137,103], HS [142]	Low-level, competitive single population ensemble with global adaptation
DE [29], PSO [146]	Low-level, cooperative multi population ensemble with uni-directional information flow and information exchange happens in every generation from explorative to exploitative population
DE [155]	Low-level, cooperative multi population ensemble where populations are sampled from the combined population every generation facilitating information exchange
DE [67]	Low-level, cooperative multi population ensemble with completely connected net topology and information exchange at regular intervals
EP [149], BBO [49,158]	Low-level, cooperative multi population ensemble where interaction happens in every generation due to offspring exchange
DE [26]	Low-level, competitive multi population ensemble
AMALGAM [32], DE [139]	High-level, competitive single population ensemble
UMOEA [125,145]	High-level, competitive multi population ensemble
Multi-objective optimization	
Pareto-based MOPSO [57]	Low-level, cooperative ensemble with multiple populations
MOEA/D [27,140]	Low-level, competitive single population ensemble
Biogeography-based optimization [51]	High-level, cooperative multi population ensemble
Multi-modal optimization	
GA [62], DE [93,150,159]	Low-level, cooperative multi population ensemble
DE [62]	Low-level, competitive single population ensemble with global adaptation
Dynamic optimization	
EP [59]	Low-level, cooperative multi population ensemble
MOEA [60]	High-level, competitive single population ensemble
Constrained optimization	
EP and DE [53], MODE [160]	Low-level, cooperative multi population ensemble
DE [54]	Low-level, competitive single population ensemble
PSO [55], GA [56]	High-level, competitive multi population ensemble
Discrete combinatorial optimization	
DE [148]	Low-level, cooperative multi population ensemble
ABC [136]	Low-level, competitive single population ensemble
Ensemble of surrogates	
Ensemble of surrogates [141]	Low-level, competitive single population ensemble with global adaptation

Table 2
Similarities and differences between ensemble-based optimization algorithms and related approaches.

Properties	Hyper-heuristics	Island Models	Adaptive operator selection	Ensemble-based
Single Population	YES	NO	YES	YES
Multiple Populations	NO	YES	NO	YES
Evolution Parallelization	NO	YES	NO	YES
Adaptive	YES	YES	YES	YES
Explicit Memory	HIGH	MODERATE	MODERATE	MODERATE
Information Exchange	NO	OCCASIONAL	NO	OCCASIONAL/FREQUENT

6. Directions of future research

Although fruitful results have been achieved in the research on ensemble strategies in POAs during the last decade, there are still some interesting open problems needed to be addressed. Here, we unfold some potential future research directions in the area of ensemble strategies for POAs.

1) Large-scale ensemble: Currently, the number of elements included in an ensemble is relatively small (e.g. around three). Now that different algorithms/search strategies/parameter values are suited to different kinds of optimization problems, it could be beneficial to make an ensemble contain larger number of elements with diversified characteristics. However, new issues may arise along with such large-scale ensemble. For example, how to implement a large-scale ensemble? The robustness of the larger scale ensemble should be investigated, since the ensemble system becomes complex and may become inefficient. In addition, the computational resource needed by a large-scale ensemble deserves to be studied, as some inappropriate elements may waste computational resources when solving a specified problem.
2) Hierarchical ensemble: Ensemble strategies in POAs can be categorized into high-level ensemble and low-level ensemble. It could be promising to combine the high-level ensemble and low-level ensemble in one POA. We may simultaneously combine multiple methods and multiple search strategies in an ensemble. In addition, it is possible to integrate multiple methods in an ensemble and these constituent methods are with the ensemble of multiple search strategies. In addition, most of the current ensemble methods are either competitive or cooperative. In hierarchical ensemble, the ensemble strategies performing exploration can be cooperative while the strategies performing exploitation can be competitive. In other words, a hierarchical ensemble that is cooperative and competitive can be designed.
3) Heterogeneous ensemble: Major ensemble POAs are the ensembles of homogeneous algorithmic components, e.g., a DE algorithm with the ensemble of multiple mutation strategies. To make the components of an ensemble more diversified, it is promising to study the heterogeneous ensemble. For example, it is possible to implement the ensemble of search strategies from differential evolution, particle swarm optimization, genetic algorithm, ant colony optimization and even some derivative based optimization approaches.
4) Ensemble of algorithmic components: The ensemble of many types of algorithmic components has been well studied, including parameter values, mutation strategies, crossover strategies, constraint handling techniques, etc. There are additional algorithmic components to be taken to be under the framework of ensemble, such as neighborhood structures, index-based topologies, Euclidean distance based topologies [204], selection approaches, and performance evaluation metrics and so on.
5) Characteristic identification of the components of an ensemble: It is generally believed that the constituent components of an ensemble should be with distinct characteristics (e.g., exploration and exploitation) to design an efficient ensemble POA. However, how to reliably determine the characteristics of different algorithmic components is still an unresolved problem. It is meaningful to design suitable metrics and methods to extract the features of algorithmic components and select appropriate ones to form an ensemble.
6) Theoretical analysis of ensemble POAs: The published literature mainly focuses on the design of appropriate mechanisms for ensemble strategies and the effectiveness of an ensemble strategy is usually verified through numerical simulations. It is necessary to analyze the ensemble POAs theoretically and describe the effects of each component of an ensemble. The knowledge about the conditions on which the ensemble strategy takes effect could guide us to design efficient ensemble POAs.
7) Effective technique for implementing the ensemble POAs: The ensemble POA is not just a mechanical combination of different algorithms or algorithmic components. It is necessary to make constituent components cooperate during the optimization process rather than purely share and compete for computational resources. In addition, due to the inherent randomness in the POAs, the existing of multiple different algorithmic components may cause the instability of the ensemble POAs (e.g. the difference among the results obtained by different runs may be significant). Therefore, it is important to propose sophisticated mechanisms to realize the ensemble POA. For example, it could be useful to comprehensively consider several factors, such as the fitness improvement, population diversity and landscape characteristic, when evaluating the components of an ensemble in the optimization process. In addition, it is also important to design efficient interaction mechanisms among different components.
8) Ensemble strategies for learning-based optimization: In learning-based optimization, the algorithm is initially expected to explore the huge database of optimization algorithms/search strategies/parameter values before settling down on a few combinations of optimization algorithms/search strategies/parameter values that suit the problem. Therefore, the idea of starting with an ensemble of huge combinations before fine tuning the ensemble to suit the problem at hand can be explored.

7. Conclusions

It is known that different optimization algorithms/search strategies/parameter values are suited to different types of optimization problems. Moreover, the most appropriate optimization algorithm/search strategy/parameter value may vary at the different stages of the optimization process. It is believed that optimization algorithms/search strategies/parameter values with different advantages and characteristics could support one another during optimization process to provide diversified search/sampling strategies for complex landscapes, thereby potentially resulting in a more efficient overall optimization algorithm. Ensemble strategy has become a promising paradigm and framework for designing versatile POAs and improving the efficiency of algorithmic configuration. This paper presents a multi-facet survey on the research related to ensemble POAs. We also provide a brief survey of other related techniques in the literature and compare them with the ensemble-based optimization algorithms. In addition, we suggest some directions for future research on additional ways of incorporating ensemble strategies into population-based optimization algorithms.

Acknowledgement

This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology under the Grant NRF-2015R1C1A1A01055669. This work was also supported by the National Natural Science Foundation of China under Grant (No. 61603404), the Natural Science Foundation of Hunan Province (No. 2017JJ3369), and the Research Foundation of National University of Defense Technology (ZK16-03-30).

Appendix A. Supplementary data

Supplementary data related to this article can be found at https://doi.org/10.1016/j.swevo.2018.08.015.

References

[1] J.H. Holland, Adaptation in Natural and Artificial Systems: an Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence, MIT press, 1992.
[2] T. Back, Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms, Oxford university press, 1996.
[3] N. Hansen, S.D. Müller, P. Koumoutsakos, Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES), Evol. Comput. 11 (2003) 1–18.
[4] D.B. Fogel, Evolutionary Computation: the Fossil Record, Wiley-IEEE Press, 1998.
[5] X. Yao, Y. Liu, G. Lin, Evolutionary programming made faster, IEEE Trans. Evol. Comput. 3 (1999) 82–102.
[6] R. Storn, K. Price, Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces, J. Global Optim. 11 (1997) 341–359.
[7] M. Dorigo, V. Maniezzo, A. Colorni, Ant system: optimization by a colony of cooperating agents, IEEE Trans. Syst. Man Cybern. Part B (Cybernetics) 26 (1996) 29–41.
[8] R. Eberhart, J. Kennedy, A new optimizer using particle swarm theory. Proceedings of the Sixth International Symposium on Micro Machine and Human Science, 1995, pp. 39–43.
[9] D. Karaboga, B. Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, J. Global Optim. 39 (2007) 459–471.
[10] L.N. De Castro, J. Timmis, Artificial Immune Systems: a New Computational Intelligence Approach, Springer Science & Business Media, 2002.
[11] G. Wu, Across neighborhood search for numerical optimization, Inf. Sci. 329 (2016) 597–618.
[12] E.K. Burke, G. Kendall, Search Methodologies, Springer, 2014.
[13] P.N. Suganthan, N. Hansen, J.J. Liang, K. Deb, Y.-P. Chen, A. Auger, et al., Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization, KanGAL Rep. 2005005 (2005) 2005.
[14] J. Liang, B. Qu, P. Suganthan, Problem Definitions and Evaluation Criteria for the CEC 2014 Special Session and Competition on Single Objective Real-parameter Numerical Optimization, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore, 2013.
[15] K. Tang, X. Yáo, P.N. Suganthan, C. MacNish, Y.-P. Chen, C.-M. Chen, et al., Benchmark functions for the CEC'2008 special session and competition on large scale global optimization, in: Nature Inspired Computation and Applications Laboratory, vol. 24, USTC, China, 2007.
[16] Q. Zhang, A. Zhou, S. Zhao, P.N. Suganthan, W. Liu, S. Tiwari, technical report, Multiobjective Optimization Test Instances for the CEC 2009 Special Session and Competition, vol. 264, University of Essex, Colchester, UK and Nanyang technological University, Singapore, 2008.
[17] R.M. Guohua Wu, P.N. Suganthan, Problem Definitions and Evaluation Criteria for the CEC 2017 Competition and Special Session on Constrained Single Objective Real-parameter Optimization, National University of Defense Technology, Kyungpook National University, Nanyang Technological University, 2016. Technical Report.
[18] S. Das, P.N. Suganthan, Problem Definitions and Evaluation Criteria for CEC 2011 Competition on Testing Evolutionary Algorithms on Real World Optimization Problems, Jadavpur University, Nanyang Technological University, Kolkata, 2010.
[19] G. Wu, J. Liu, M. Ma, D. Qiu, A two-phase scheduling method with the consideration of task clustering for earth observing satellites, Comput. Oper. Res. 40 (2013) 1884–1894.
[20] Y. Bazi, N. Alajlan, F. Melgani, H. Alhichri, S. Malek, R.R. Yager, Differential evolution extreme learning machine for the classification of hyperspectral images, Geosci. Rem. Sens. Lett. IEEE 11 (2014) 1066–1070.
[21] T. Bhadra, S. Bandyopadhyay, Unsupervised feature selection using an improved version of Differential Evolution, Expert Syst. Appl. 42 (2015) 4042–4053.
[22] G. Wu, W. Pedrycz, H. Li, M. Ma, J. Liu, Coordinated planning of heterogeneous earth observation resources, IEEE Trans. Syst. Man Cybern.: Systems 46 (2016) 109–125.
[23] Y. Chen, V. Mahalec, Y. Chen, X. Liu, R. He, K. Sun, Reconfiguration of satellite orbit for cooperative observation using variable-size multi-objective differential evolution, Eur. J. Oper. Res. 242 (2015) 10–20.
[24] Y.Z. Hsieh, M.C. Su, P.C. Wang, A PSO-based rule extractor for medical diagnosis, J. Biomed. Inf. 49 (2014) 53–60.
[25] W. Gong, Á. Fialho, Z. Cai, H. Li, Adaptive strategy selection in differential evolution for numerical optimization: an empirical study, Inf. Sci. 181 (2011) 5364–5386.
[26] G. Wu, R. Mallipeddi, P.N. Suganthan, R. Wang, H. Chen, Differential evolution with multi-population based ensemble of mutation strategies, Inf. Sci. 329 (2016) 329–345.
[27] S.-Z. Zhao, P.N. Suganthan, Q. Zhang, Decomposition-based multiobjective evolutionary algorithm with an ensemble of neighborhood sizes, IEEE Trans. Evol. Comput. 16 (2012) 442–446.
[28] R. Mallipeddi, P.N. Suganthan, Q.-K. Pan, M.F. Tasgetiren, Differential evolution algorithm with ensemble of parameters and mutation strategies, Appl. Soft Comput. 11 (2011) 1679–1696.
[29] N. Lynn, R. Mallipeddi, P.N. Suganthan, Differential evolution with two subpopulations, in: International Conference on Swarm, Evolutionary, and Memetic Computing, Springer International Publishing, 2015, pp. 1–13.
[30] Y. Ren, L. Zhang, P.N. Suganthan, Ensemble classification and regression-recent developments, applications and future directions, IEEE Comput. Intell. Mag. 11 (2016) 41–53.
[31] S. Zhou, Z. Sun, Can ensemble method convert a 'weak' evolutionary algorithm to a 'Strong' one?, in: International Conference on Computational Intelligence for Modelling, Control and Automation, and International Conference on Intelligent Agents, Web Technologies and Internet Commerce, 2005, pp. 68–74.
[32] J.A. Vrugt, B.A. Robinson, J.M. Hyman, Self-adaptive multimethod search for global optimization in real-parameter spaces, IEEE Trans. Evol. Comput. 13 (2009) 243–259.
[33] F. Peng, K. Tang, G. Chen, X. Yao, Population-based algorithm portfolios for numerical optimization, IEEE Trans. Evol. Comput. 14 (2010) 782–800.
[34] E.K. Burke, M. Gendreau, M. Hyde, G. Kendall, G. Ochoa, E. Özcan, et al., Hyper-heuristics: a survey of the state of the art, J. Oper. Res. Soc. 64 (2013) 1695–1724.
[35] X. Chen, Y.S. Ong, M.H. Lim, K.C. Tan, A multi-facet survey on memetic computation, IEEE Trans. Evol. Comput. 15 (2011) 591–607.
[36] C. Grosan, A. Abraham, Hybrid evolutionary algorithms: methodologies, architectures, and reviews, in: Hybrid Evolutionary Algorithms, Springer, 2007, pp. 1–17.
[37] P. Moscato, Caltech concurrent computation program, C3P Report, On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts: Towards Memetic Algorithms, vol. 826, 1989, p. 1989.
[38] G. Wu, D. Qiu, Y. Yu, W. Pedrycz, M. Ma, H. Li, Superior solution guided particle swarm optimization combined with local search techniques, Expert Syst. Appl. 41 (2014) 7536–7548, 2014/11/15.
[39] K. Tang, F. Peng, G. Chen, X. Yao, Population-based Algorithm Portfolios with automated constituent algorithms selection, Inf. Sci. 279 (2014) 94–104, 2014/09/20.
[40] E.K. Burke, M. Hyde, G. Kendall, G. Ochoa, E. Özcan, J.R. Woodward, A classification of hyper-heuristic approaches, in: Handbook of Metaheuristics, Springer, 2010, pp. 449–468.
[41] A.K. Qin, V.L. Huang, P.N. Suganthan, Differential evolution algorithm with strategy adaptation for global numerical optimization, IEEE Trans. Evol. Comput. 13 (2009) 398–417.
[42] R.A. Sarker, S.M. Elsayed, T. Ray, Differential evolution with dynamic parameters selection for optimization problems, IEEE Trans. Evol. Comput. 18 (2014) 689–707.
[43] W. Du, B. Li, Multi-strategy ensemble particle swarm optimization for dynamic optimization, Inf. Sci. 178 (2008) 3096–3109.
[44] Y. Wang, B. Li, T. Weise, J. Wang, B. Yuan, Q. Tian, Self-adaptive learning based particle swarm optimization, Inf. Sci. 181 (2011) 4515–4538.
[45] C. Li, S. Yang, T.T. Nguyen, A self-learning particle swarm optimizer for global optimization problems, IEEE Trans. Syst. Man Cybern. Part B (Cybernetics) 42 (2012) 627–646.
[46] H. Wang, Z. Wu, S. Rahnamayan, H. Sun, Y. Liu, J.S. Pan, Multi-strategy ensemble artificial bee colony algorithm, Inf. Sci. 279 (2014) 587–603.
[47] H. Wang, W. Wang, H. Sun, Multi-strategy ensemble artificial bee colony algorithm for large-scale production scheduling problem, Int. J. Innovat. Comput. Appl. 6 (2015) 128.
[48] X. Zhou, M. Wang, J. Wan, J. Zuo, An improved multi-strategy ensemble artificial bee colony algorithm with neighborhood search, in: International Conference on Neural Information Processing, 2016, pp. 489–496.
[49] G. Xiong, D. Shi, X. Duan, Multi-strategy ensemble biogeography-based optimization for economic dispatch problems, Appl. Energy 111 (2013) 801–811.
[50] H. Ma, M. Fei, Z. Ding, J. Jin, Biogeography-based optimization with ensemble of migration models for global numerical optimization, in: Evolutionary Computation (CEC), 2012 IEEE Congress on, 2012, pp. 1–8.
[51] H. Ma, S. Su, D. Simon, M. Fei, Ensemble multi-objective biogeography-based optimization with application to automated warehouse scheduling, Eng. Appl. Artif. Intell. 44 (Sep 2015) 79–90.
[52] W. Gong, A. Zhou, Z. Cai, A multioperator search strategy based on cheap surrogate models for evolutionary optimization, IEEE Trans. Evol. Comput. 19 (2015) 746–758.
[53] R. Mallipeddi, P.N. Suganthan, Ensemble of constraint handling techniques, IEEE Trans. Evol. Comput. 14 (2010) 561–579.

[54] S.M. Elsayed, R.A. Sarker, D.L. Essam, A self-adaptive combined strategies algorithm for constrained optimization using differential evolution, Appl. Math. Comput. 241 (2014) 267–282, 2014/08/15.
[55] S.M. Elsayed, R.A. Sarker, E. Mezura-Montes, Self-adaptive mix of particle swarm methodologies for constrained optimization, Inf. Sci. 277 (2014) 216–233, 2014/09/01.
[56] S.M. Elsayed, R.A. Sarker, D.L. Essam, Training and testing a self-adaptive multi-operator evolutionary algorithm for constrained optimization, Appl. Soft Comput. 26 (2015) 515–522, 2015/01/01.
[57] S.-Z. Zhao, P.N. Suganthan, Multi-objective evolutionary algorithm with ensemble of external archives, Int. J. Innov. Comput. Inf. Control 6 (April 2010) 1713–1726.
[58] G.G. Yen, H. Zhenan, Performance metric ensemble for multiobjective evolutionary algorithms, IEEE Trans. Evol. Comput. 18 (2014) 131–144.
[59] E.L. Yu, P.N. Suganthan, Evolutionary programming with ensemble of explicit memories for dynamic optimization, in: IEEE Congress on Evolutionary Computation, 2009, pp. 431–438.
[60] Y. Wang, B. Li, Multi-strategy ensemble evolutionary algorithm for dynamic multi-objective optimization, Memetic Comput. 2 (2010) 3–24.
[61] B.-Y. Qu, P.N. Suganthan, J.J. Liang, Differential evolution with neighborhood mutation for multimodal optimization, IEEE Trans. Evol. Comput. 16 (2012) 601–614.
[62] E.L. Yu, P.N. Suganthan, Ensemble of niching algorithms, Inf. Sci. 180 (2010) 2815–2833.
[63] S. Hui, P.N. Suganthan, Ensemble and arithmetic recombination-based speciation differential evolution for multimodal optimization, IEEE Trans. Cybern. 46 (2016) 64–74.
[64] Q. Fan, X. Yan, Differential evolution algorithm with self-adaptive strategy and control parameters for P-xylene oxidation process optimization, Soft Comput. 19 (2015) 1363–1391.
[65] T. Takahama, S. Sakai, Differential evolution with dynamic strategy and parameter selection by detecting landscape modality, in: Congress on Evolutionary Computation, 2012, pp. 1–8.
[66] Y. Wang, Z. Cai, Q. Zhang, Differential evolution with composite trial vector generation strategies and control parameters, IEEE Trans. Evol. Comput. 15 (2011) 55–66.
[67] M.Z. Ali, N.H. Awad, P.N. Suganthan, Multi-population differential evolution with balanced ensemble of mutation strategies for large-scale global optimization, Appl. Soft Comput. 33 (2015) 304–327.
[68] X. Li, S. Ma, J. Hu, Multi-search differential evolution algorithm, Appl. Intell. (2017) 1–26.
[69] S.C. Wang, Differential evolution optimization with time-frame strategy adaptation, Soft Comput. (2016) 1–22.
[70] Q. Lin, C. Tang, Y. Ma, Z. Du, J. Li, J. Chen, et al., A novel adaptive control strategy for decomposition-based multiobjective algorithm, Comput. Oper. Res. 78 (2016) 94–107.
[71] Q.Z. Lin, Z.W. Liu, Q. Yan, Z.H. Du, C.A.C. Coello, Z.P. Liang, et al., Adaptive composite operator selection and parameter control for multiobjective evolutionary algorithm, Inf. Sci. 339 (Apr 2016) 332–352.
[72] L.Z. Cui, G.H. Li, Q.Z. Lin, J.Y. Chen, N. Lu, Adaptive differential evolution algorithm with novel mutation strategies in multiple sub-populations, Comput. Oper. Res. 67 (Mar 2016) 155–173.
[73] X.T. Li, S.J. Ma, Y.H. Wang, Multi-population based ensemble mutation method for single objective bilevel optimization problem, IEEE Access. 4 (2016).
[74] R. Polakova, J. Tvrdik, P. Bujok, IEEE, Evaluating the performance of L-SHADE with competing strategies on CEC2014 single parameter-operator test suite, in: 2016 Ieee Congress on Evolutionary Computation (Cec), 2016, pp. 1181–1187.
[75] M. Paldrak, M.F. Tasgetiren, P.N. Suganthan, Q.K. Pan, An ensemble of differential evolution algorithms with variable neighborhood search for constrained function optimization, in: 2016 Ieee Congress on Evolutionary Computation (Cec), 2016, pp. 2610–2617.
[76] E. Debie, S.M. Elsayed, D.L. Essam, R.A. Sarker, Investigating multi-operator differential evolution for feature selection, in: T. Ray, R. Sarker, X. Li (Eds.), Artificial Life and Computational Intelligence, Acalci 2016, vol. 9592, 2016, pp. 273–284.
[77] Q.Q. Fan, X.F. Yan, Self-adaptive differential evolution algorithm with zoning evolution of control parameters and adaptive mutation strategies, IEEE Trans. Cybern. 46 (Jan 2016) 219–232.
[78] W. Yi, L. Gao, X. Li, Y. Zhou, A new differential evolution algorithm with a hybrid mutation operator and self-adapting control parameters for global optimization problems, Appl. Intell. 42 (Jun 2015) 642–660.
[79] Q.Q. Fan, Y.L. Zhang, Self-adaptive differential evolution algorithm with crossover strategies adaptation and its application in parameter estimation, Chemometr. Intell. Lab. Syst. 151 (Feb 2016) 164–171.
[80] R. Polakova, J. Tvrdik, P. Bujok, IEEE, L-SHADE with competing strategies applied to CEC2015 learning-based test suite, in: 2016 Ieee Congress on Evolutionary Computation (Cec), 2016, pp. 4790–4796.
[81] H. Rakhshani, A. Rahati, Intelligent multiple search strategy cuckoo algorithm for numerical and engineering optimization problems, Arabian J. Sci. Eng. 42 (Feb 2017) 567–593.
[82] C. Wang, Y.C. Liu, Y. Chen, Y. Wei, Self-adapting hybrid strategy particle swarm optimization algorithm, Soft Comput. 20 (Dec 2016) 4933–4963.
[83] H. Samma, C.P. Lim, J.M. Saleh, A new reinforcement learning-based memetic particle swarm optimizer, Appl. Soft Comput. 43 (Jun 2016) 276–297.
[84] G. Iacca, F. Caraffini, F. Neri, Multi-strategy coevolving aging particle optimization, Int. J. Neural Syst. 24 (Feb 2014).
[85] J. Gou, W.-P. Guo, C. Wang, W. Luo, A multi-strategy improved particle swarm optimization algorithm and its application to identifying uncorrelated multi-source load in the frequency domain, Neural Comput. Appl. 28 (Jul 2017) 1635–1656.
[86] G. Lin, K. Zhao, Q. Wan, Takagi-sugeno fuzzy model identification using coevolution particle swarm optimization with multi-strategy, Appl. Intell. 45 (Jul 2016) 187–197.
[87] K. Tang, Z. Li, L. Luo, B. Liu, Multi-strategy adaptive particle swarm optimization for numerical optimization, Eng. Appl. Artif. Intell. 37 (Jan 2015) 9–19.
[88] H. Xia, J. Zhuang, D. Yu, Combining crowding estimation in objective and decision space with multiple selection and search strategies for multi-objective evolutionary optimization, IEEE Trans. Cybern. 44 (Mar 2014) 378–393.
[89] Y. Li, A. Zhou, G. Zhang, Ieee, An MOEA/D with multiple differential evolution mutation operators, in: 2014 Ieee Congress on Evolutionary Computation (Cec), 2014, pp. 397–404.
[90] G. Karafotias, M. Hoogendoorn, Á.E. Eiben, Parameter control in evolutionary algorithms: trends and challenges, IEEE Trans. Evol. Comput. 19 (2015) 167–187.
[91] X.-F. Liu, Z.-H. Zhan, J. Zhang, Dichotomy guided based parameter adaptation for differential evolution, in: Gecco'15: Proceedings of the 2015 Genetic and Evolutionary Computation Conference, 2015, pp. 289–296.
[92] S.M. Elsayed, R.A. Sarker, T. Ray, IEEE, Parameters adaptation in differential evolution, in: 2012 Ieee Congress on Evolutionary Computation (Cec), 2012, 2012.
[93] B.Y. Qu, P. Gouthanan, P.N. Suganthan, Dynamic grouping crowding differential evolution with ensemble of parameters for multi-modal optimization, in: B.K. Panigrahi, S. Das, P.N. Suganthan, S.S. Dash (Eds.), Swarm, Evolutionary, and Memetic Computing, vol. 6466, 2010, p. 19.
[94] G. Iacca, F. Neri, F. Caraffini, P.N. Suganthan, A differential evolution framework with ensemble of parameters and strategies and pool of local search algorithms, in: A.I. EsparciaAlcazar (Ed.), Applications of Evolutionary Computation, vol. 8602, 2014, pp. 615–626.
[95] J. Zhang, A.C. Sanderson, JADE: adaptive differential evolution with optional external archive, IEEE Trans. Evol. Comput. 13 (2009) 945–958.
[96] R. Tanabe, A.S. Fukunaga, Improving the search performance of SHADE using linear population size reduction, in: 2014 IEEE Congress on Evolutionary Computation (CEC), 2014, pp. 1658–1665.
[97] R. Tanabe, A. Fukunaga, Success-history based parameter adaptation for Differential Evolution, in: 2013 IEEE Congress on Evolutionary Computation, 2013, pp. 71–78.
[98] Z.H. Zhan, J. Zhang, Y. Li, H.S.H. Chung, Adaptive particle swarm optimization, IEEE Trans. Syst. Man Cybern. Part B (Cybernetics) 39 (2009) 1362–1381.
[99] J. Brest, S. Greiner, B. Boskovic, M. Mernik, V. Zumer, Self-Adapting control parameters in differential evolution: a comparative study on numerical benchmark problems, IEEE Trans. Evol. Comput. 10 (2006) 646–657.
[100] M. Krithikaa, R. Mallipeddi, IEEE, Differential evolution with an ensemble of low-quality surrogates for expensive optimization problems, in: 2016 Ieee Congress on Evolutionary Computation (Cec), 2016, pp. 78–85.
[101] C. Jin, A.K. Qin, K. Tang, Ieee, Local ensemble surrogate assisted crowding differential evolution, in: 2015 Ieee Congress on Evolutionary Computation (Cec), 2015, pp. 433–440.
[102] P. Karthikeyan, S. Baskar, Genetic algorithm with ensemble of immigrant strategies for multicast routing in Ad hoc networks, Soft Comput. 19 (2015) 489–498.
[103] N.H. Awad, M.Z. Ali, P.N. Suganthan, R.G. Reynolds, An ensemble sinusoidal parameter adaptation incorporated with L-SHADE for solving CEC2014 benchmark problems, in: IEEE Congress on Evolutionary Computation, 2016, pp. 2958–2965.
[104] Y. Xue, S. Zhong, Y. Zhuang, B. Xu, An ensemble algorithm with self-adaptive learning techniques for high-dimensional numerical optimization, Appl. Math. Comput. 231 (2014) 329–346.
[105] S.M. Elsayed, R.A. Sarker, D.L. Essam, Self-adaptive differential evolution incorporating a heuristic mixing of operators, Comput. Optim. Appl. 54 (Apr 2013) 771–790.
[106] J. Brest, B. Bošković, A. Zamuda, I. Fister, and E. Mezura-Montes, "Real Parameter Single Objective Optimization Using Self-adaptive Differential Evolution Algorithm with More Strategies," pp. 377-383.
[107] J.-H. Zhong, M. Shen, J. Zhang, H.S.-H. Chung, Y.-H. Shi, Y. Li, A differential evolution algorithm with dual populations for solving periodic railway timetable scheduling problem, IEEE Trans. Evol. Comput. 17 (Aug 2013) 512–527.
[108] A.P. Piotrowski, Adaptive memetic differential evolution with global and local neighborhood-based mutation operators, Inf. Sci. 241 (Aug 20 2013) 164–194.
[109] J. Cheng, G. Zhang, F. Caraffini, F. Neri, Multicriteria adaptive differential evolution for global numerical optimization, Integrated Comput. Aided Eng. 22 (2015) 103–117.
[110] T. J. Choi and C. W. Ahn, "An Adaptive Cauchy Differential Evolution Algorithm with Population Size Reduction and Modified Multiple Mutation Strategies," pp. 13-26.
[111] W.-l. Xiang, X.-l. Meng, M.-q. An, Y.-z. Li, M.-x. Gao, An enhanced differential evolution algorithm based on multiple mutation strategies, Comput. Intell. Neurosci. 2015 (2015).
[112] N.R. Sabar, S.Y. Chong, G. Kendall, A Hybrid Differential Evolution Algorithm - Game Theory for the Berth Allocation Problem, 2015.
[113] X. Lu, K. Tang, B. Sendhoff, X. Yao, A new self-adaptation scheme for differential evolution, Neurocomputing 146 (Dec 25 2014) 2–16.
[114] L. Tang, Y. Dong, J. Liu, Differential evolution with an individual-dependent mechanism, IEEE Trans. Evol. Comput. 19 (Aug 2015) 560–574.

[115] X.Y. Zhou, Z.J. Wu, H. Wang, S. Rahnamayan, Enhancing differential evolution with role assignment scheme, Soft Comput. 18 (Nov 2014) 2209–2225.
[116] S.X. Zhang, S.Y. Zheng, L.M. Zheng, An efficient multiple variants coordination framework for differential evolution, IEEE Trans. Cybern. 47 (2017) 2780–2793.
[117] G.H. Li, Q.Z. Lin, L.Z. Cui, Z.H. Du, Z.P. Liang, J.Y. Chen, et al., A novel hybrid differential evolution algorithm with modified CoDE and JADE, Appl. Soft Comput. 47 (Oct 2016) 577–599.
[118] S. Thangavelu, C.S. Velayutham, An investigation on mixing heterogeneous differential evolution variants in a distributed framework, Int. J. Bio-Inspired Comput. 7 (2015 2015) 307–320.
[119] G. Wu, X. Shen, H. Li, H. Chen, A. Lin, P.N. Suganthan, Ensemble of differential evolution variants, Inf. Sci. 423 (1//2018) 172–186.
[120] N. Lynn, P.N. Suganthan, Ensemble particle swarm optimizer, Appl. Soft Comput. 55 (2017) 533–548.
[121] S.Y. Yuen, X. Zhang, On composing an algorithm portfolio, Memetic Comput. 7 (Sep 2015) 203–214.
[122] J. Grobler, A.P. Engelbrecht, G. Kendall, V.S.S. Yadavalli, IEEE, in: Multi-method Algorithms: Investigating the Entity-to-algorithm Allocation Problem, 2013.
[123] S.M. Elsayed, R.A. Sarker, D.L. Essam, Adaptive Configuration of evolutionary algorithms for constrained optimization, Appl. Math. Comput. 222 (Oct 1 2013) 680–711.
[124] S. Elsayed, N. Hamza, R. Sarker, Testing United Multi-operator Evolutionary Algorithms-ii on Single Objective Optimization Problems, 2016, pp. 2966–2973.
[125] S.M. Elsayed, R.A. Sarker, D.L. Essam, N.M. Hamza, Testing united multi-operator evolutionary algorithms on the CEC2014 real-parameter numerical optimization, in: IEEE Congress on Evolutionary Computation, 2014, pp. 1650–1657.
[126] S.Y. Yuen, X. Zhang, Ieee, Multiobjective evolutionary algorithm portfolio: choosing suitable algorithm for multiobjective optimization problem, in: 2014 Ieee Congress on Evolutionary Computation (Cec), 2014, pp. 1967–1973.
[127] L. Hongxin, L. Yinhong, C. Jinfu, Adaptive multiple evolutionary algorithms search for multi-objective optimal reactive power dispatch, Int. Trans. Electr. Energy Syst. 24 (2014) 780–795.
[128] D.T. Xie, L.X. Ding, Y.R. Hu, S.W. Wang, C.W. Xie, L. Jiang, A multi-algorithm balancing convergence and diversity for multi-objective optimization, J. Inf. Sci. Eng. 29 (Sep 2013) 811–834.
[129] X. Zhang, R. Srinivasan, M.V. Liew, On the use of multi-algorithm, genetically adaptive multi-objective method for multi-site calibration of the SWAT model, Hydrol. Process. 24 (2010) 955–969.
[130] J.A. Vrugt, B.A. Robinson, Improved evolutionary optimization from genetically adaptive multimethod search, Proc. Natl. Acad. Sci. Unit. States Am. 104 (2007) 708–711.
[131] W.K. Mashwani, A. Salhi, Multiobjective evolutionary algorithm based on multimethod with dynamic resources allocation, Appl. Soft Comput. 39 (2016) 292–309, 2016/02/01.
[132] P. Baudiš, P. Pošík, Online Black-box Algorithm Portfolios for Continuous Optimization, 2014.
[133] R. Akay, A. Basturk, A. Kalinli, X. Yao, Parallel population-based algorithm portfolios: an empirical study, Neurocomputing 247 (2017/07/19/2017) 115–125.
[134] W.M. Spears, Adapting crossover in evolutionary algorithms, in: 4th Annual Conference on Evolutionary Programming, 1995, pp. 367–384.
[135] R. Mallipeddi, P.N. Suganthan, Differential evolution algorithm with ensemble of populations for global numerical optimization, OPSEARCH 46 (2009) 184–213.
[136] K.-Z. Gao, P.N. Suganthan, T.J. Chua, C.S. Chong, T.X. Cai, Q.-K. Pan, A two-stage artificial bee colony algorithm scheduling flexible job-shop scheduling problem with new job insertion, Expert Syst. Appl. 42 (2015) 7652–7663.
[137] N. Lynn, R. Mallipeddi, P.N. Suganthan, Self-adaptive ensemble differential evolution with sampled parameter values for unit commitment, in: International Conference on Swarm, Evolutionary, and Memetic Computing, Springer International Publishing, 2016, pp. 1–16.
[138] H.-S. Yoon, B.-R. Moon, An empirical study on the synergy of multiple crossover operators, IEEE Trans. Evol. Comput. 6 (2002) 212–223.
[139] A. LaTorre, S. Muelas, J.-M. Peña, A MOS-based dynamic memetic differential evolution algorithm for continuous optimization: a scalability test, Soft Comput. 15 (2011) 2187–2199.
[140] W. Khan, Q. Zhang, MOEA/D-DRA with two crossover operators, in: UK Workshop on Computational Intelligence, 2010, pp. 1–6.
[141] T. Goel, R.T. Haftka, W. Shyy, N.V. Queipo, Ensemble of surrogates, Struct. Multidiscip. Optim. 33 (2007) 199–216.
[142] Q.-K. Pan, P.N. Suganthan, M.F. Tasgetiren, A harmony search algorithm with ensemble of parameter sets, in: IEEE Congress on Evolutionary Computation, 2009, pp. 1815–1820.
[143] R. Mallipeddi, M. Lee, An evolving surrogate model-based differential evolution algorithm, Appl. Soft Comput. 34 (2015) 770–787.
[144] S.M. Elsayed, R.A. Sarker, D.L. Essam, Multi-operator based evolutionary algorithms for solving constrained optimization problems, Comput. Oper. Res. 38 (2011) 1877–1896.
[145] S.M. Elsayed, R.A. Sarker, D.L. Essam, United multi-operator evolutionary algorithms, in: IEEE Congress on Evolutionary Computation, 2014, pp. 1006–1013.
[146] N. Lynn, P.N. Suganthan, Heterogeneous comprehensive learning particle swarm optimization with enhanced exploration and exploitation, Swarm Evol. Comput. 24 (2015) 11–24.
[147] R. Mallipeddi, G. Iacca, P.N. Suganthan, F. Neri, E. Mininno, Ensemble strategies in compact differential evolution, in: IEEE Congress on Evolutionary Computation, 2011, pp. 1972–1977.
[148] M.F. Tasgetiren, P.N. Suganthan, Q.-K. Pan, An ensemble of discrete differential evolution algorithms for solving the generalized traveling salesman problem, Appl. Math. Comput. 215 (2010) 3356–3368.
[149] R. Mallipeddi, S. Mallipeddi, P.N. Suganthan, Ensemble strategies with adaptive evolutionary programming, Inf. Sci. 180 (2010) 1571–1581.
[150] B.-Y. Qu, P.N. Suganthan, Novel multimodal problems and differential evolution with ensemble of restricted tournament selection, in: IEEE Congress on Evolutionary Computation, 2010, pp. 1–7.
[151] R. Mallipeddi, P.N. Suganthan, Differential evolution algorithm with ensemble of parameters and mutation and crossover strategies, in: Swarm, Evolutionary, and Memetic Computing, vol. 6466, Springer Berlin Heidelberg, 2010, pp. 71–78.
[152] Q.-K. Pan, P.N. Suganthan, L. Wang, L. Gao, R. Mallipeddi, A differential evolution algorithm with self-adapting strategy and control parameters, Comput. Oper. Res. 38 (2011) 394–408.
[153] A. Zamuda, J. Brest, Self-adaptive control parameters' randomization frequency and propagations in differential evolution, Swarm Evol. Comput. 25 (12//2015) 72–99.
[154] A. Zamuda, J. Brest, E. Mezura-Montes, Structured population size reduction differential evolution with multiple mutation strategies on CEC 2013 real parameter optimization, in: 2013 IEEE Congress on Evolutionary Computation, 2013, pp. 1925–1931.
[155] X. Li, S. Ma, Y. Wang, Multi-population based ensemble mutation method for single objective bilevel optimization problem, IEEE Access. 4 (2016) 7262–7274.
[156] X. Zhou, M. Wang, J. Wan, J. Zuo, An improved multi-strategy ensemble artificial bee colony algorithm with neighborhood search, in: A. Hirose, S. Ozawa, K. Doya, K. Ikeda, M. Lee, D. Liu (Eds.), Neural Information Processing: 23rd International Conference, ICONIP 2016, Kyoto, Japan, October 16–21, 2016, Proceedings, Part IV, Springer International Publishing, Cham, 2016, pp. 489–496.
[157] H. Wang, W. Wang, H. Sun, Multi-strategy ensemble artificial bee colony algorithm for large-scale production scheduling problem, Int. J. Innovat. Comput. Appl. 6 (2015) 128–136, 2015/01/01.
[158] M. Haiping, F. Minrui, D. Zhiguo, J. Jing, Biogeography-based optimization with ensemble of migration models for global numerical optimization, in: 2012 IEEE Congress on Evolutionary Computation, 2012, pp. 1–8.
[159] B.-Y. Qu, J. Liang, P.N. Suganthan, T. Chen, Ensemble of clearing differential evolution for multi-modal optimization, in: Advances in Swarm Intelligence, vol. 7331, Springer Berlin Heidelberg, 2012, pp. 350–357.
[160] B.-Y. Qu, P.N. Suganthan, Constrained multi-objective optimization algorithm with an ensemble of constraint handling methods, Eng. Optim. 43 (2010) 403–416.
[161] J.-Q. Li, Q.-K. Pan, K.-Z. Gao, Pareto-based discrete artificial bee colony algorithm for multi-objective flexible job shop scheduling problems, Int. J. Adv. Manuf. Technol. 55 (2011) 1159–1169.
[162] K.-Z. Gao, P.N. Suganthan, M.F. Tasgetiren, Q.K. Pan, Q.Q. Sun, Effective ensembles of heuristics for scheduling flexible job shop problem with new job insertion, Comput. Ind. Eng. 90 (2015) 107–117.
[163] R. Jin, W. Chen, T.W. Simpson, Comparative studies of metamodelling techniques under multiple modelling criteria, Struct. Multidiscip. Optim. 23 (2001) 1–13.
[164] D. Lim, Y.-S. Ong, Y. Jin, B. Sendhoff, A study on metamodeling techniques, ensembles, and multi-surrogates in evolutionary computation, in: Presented at the Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation, London, England, 2007.
[165] E.K. Burke, M. Hyde, G. Kendall, J. Woodward, A genetic programming hyper-heuristic approach for evolving 2-D strip packing heuristics, IEEE Trans. Evol. Comput. 14 (2010) 942–958.
[166] N. Pillay, A review of hyper-heuristics for educational timetabling, Ann. Oper. Res. 239 (2016) 3–38.
[167] C.-W. Tsai, W.-C. Huang, M.-H. Chiang, M.-C. Chiang, C.-S. Yang, A hyper-heuristic scheduling algorithm for cloud, IEEE Trans. Cloud Comput. 2 (2014) 236–250.
[168] S. Nguyen, M. Zhang, M. Johnston, K.C. Tan, Automatic programming via iterated local search for dynamic job shop scheduling, IEEE Trans. Cybern. 45 (2015) 1–14.
[169] E.K. Burke, S. Petrovic, R. Qu, Case-based heuristic selection for timetabling problems, J. Sched. 9 (2006) 115–132.
[170] E.K. Burke, B. McCollum, A. Meisels, S. Petrovic, R. Qu, A graph-based hyper-heuristic for educational timetabling problems, Eur. J. Oper. Res. 176 (2007) 177–192.
[171] E.K. Burke, G. Kendall, E. Soubeiga, A tabu-search hyperheuristic for timetabling and rostering, J. Heuristics 9 (2003) 451–470.
[172] K.A. Dowsland, E. Soubeiga, E.K. Burke, A simulated annealing based hyperheuristic for determining shipper sizes for storage and transportation, Eur. J. Oper. Res. 179 (2007) 759–774.
[173] Y.-J. Zheng, M.-X. Zhang, H.-F. Ling, S.-Y. Chen, Emergency railway transportation planning using a hyper-heuristic approach, IEEE Trans. Intell. Transport. Syst. 16 (2015) 321–329.
[174] A. Fukunaga, Automated discovery of local search heuristics for satisfiability testing, Evol. Comput. 16 (2008) 31–61.
[175] Z. Ren, H. Jiang, J. Xuan, Z. Luo, Hyper-heuristics with low level parameter adaptation, Evol. Comput. 20 (2012) 189–227.
[176] C. Geiger, R. Uzsoy, H. Aytuğ, Rapid modeling and discovery of priority dispatching rules: an autonomous learning approach, J. Sched. 9 (2006) 7–34.
[177] R. Bai, E.K. Burke, G. Kendall, L. Jingpeng, B. McCollum, A hybrid evolutionary approach to the nurse rostering problem, IEEE Trans. Evol. Comput. 14 (2010) 580–590.

[178] E.K. Burke, M.R. Hyde, G. Kendall, J. Woodward, Automating the packing heuristic design process with genetic programming, Evol. Comput. 20 (2012) 63–89.
[179] N.R. Sabar, M. Ayob, G. Kendall, R. Qu, Grammatical evolution hyper-heuristic for combinatorial optimization problems, IEEE Trans. Evol. Comput. 17 (2013) 840–861.
[180] B. Koohestani, R. Poli, Evolving an improved algorithm for envelope reduction using a hyper-heuristic approach, IEEE Trans. Evol. Comput. 18 (2014) 543–558.
[181] B.A. Huberman, R.M. Lukose, T. Hogg, An economics approach to hard computational problems, Science 275 (1997) 51–54.
[182] Z.M. Skolicki, An Analysis of Island Models in Evolutionary Computation, George Mason University, 2007.
[183] Z. Skolicki, K. De Jong, Improving evolutionary algorithms with multi-representation island models, in: Parallel Problem Solving from Nature - PPSN VIII, vol. 3242, Springer Berlin Heidelberg, 2004, pp. 420–429.
[184] J. Lassig, D. Sudholt, General upper bounds on the runtime of parallel evolutionary algorithms, Evol. Comput. 22 (2014) 405–437.
[185] R. Nogueras, C. Cotta, Studying fault-tolerance in island-based evolutionary and multimemetic algorithms, J. Grid Comput. (2015/01/10 2015) 1–24.
[186] J. Lässig, D. Sudholt, Analysis of speedups in parallel evolutionary algorithms for combinatorial optimization, in: T. Asano, S.-i. Nakano, Y. Okamoto, O. Watanabe (Eds.), Algorithms and Computation, vol. 7074, Springer Berlin Heidelberg, 2011, pp. 405–414.
[187] L. Araujo, J.J. Merelo, Diversity through multiculturality: assessing migrant choice policies in an island model, IEEE Trans. Evol. Comput. 15 (2011) 456–469.
[188] K. Li, A. Fialho, S. Kwong, Q. Zhang, Adaptive operator selection with bandits for a multiobjective evolutionary algorithm based on decomposition, IEEE Trans. Evol. Comput. 18 (Feb 2014) 114–130.
[189] Á. Fialho, L. Da Costa, M. Schoenauer, M. Sebag, Extreme value based adaptive operator selection, in: Parallel Problem Solving from Nature – PPSN X, vol. 5199, Springer Berlin Heidelberg, 2008, pp. 175–184.
[190] Á. Fialho, L. Da Costa, M. Schoenauer, M. Sebag, Dynamic multi-armed bandits and extreme value-based rewards for adaptive operator selection in evolutionary algorithms, in: Learning and Intelligent Optimization, vol. 5851, Springer Berlin Heidelberg, 2009, pp. 176–190.
[191] J. Maturana, Á. Fialho, F. Saubion, M. Schoenauer, M. Sebag, Extreme compass and dynamic multi-armed bandits for adaptive operator selection, in: IEEE Congress on Evolutionary Computation, 2009, pp. 365–372.
[192] Á. Fialho, M. Schoenauer, M. Sebag, Toward comparison-based adaptive operator selection, in: Presented at the Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation, Portland, Oregon, USA, 2010.
[193] D. Thierens, Adaptive strategies for operator allocation, in: Parameter Setting in Evolutionary Algorithms, vol. 54, Springer Berlin Heidelberg, 2007, pp. 77–90.
[194] P. Auer, N. Cesa-Bianchi, P. Fischer, Finite-time analysis of the multiarmed bandit problem, Mach. Learn. 47 (2002) 235–256.
[195] L. DaCosta, Á. Fialho, M. Schoenauer, M. Sebag, Adaptive operator selection with dynamic multi-armed bandits, in: Presented at the Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation, Atlanta, GA, USA, 2008.
[196] Á. Fialho, L. Da Costa, M. Schoenauer, M. Sebag, Analyzing bandit-based adaptive operator selection mechanisms, Ann. Math. Artif. Intell. 60 (2010) 25–64.
[197] J. Brest, A. Zamuda, B. Boskovic, M.S. Maucec, V. Zumer, Dynamic optimization using self-adaptive differential evolution, in: IEEE Congress on Evolutionary Computation, 2009, pp. 415–422.
[198] J. Zhang, A.C. Sanderson, Self-adaptive multi-objective differential evolution with direction information provided by archived inferior solutions, in: IEEE Congress on Evolutionary Computation, 2008, pp. 2801–2810.
[199] Y.-N. Wang, L.-H. Wu, X.-F. Yuan, Multi-objective self-adaptive differential evolution with elitist archive and crowding entropy-based diversity measure, Soft Comput. 14 (2010) 193–209.
[200] V.L. Huang, S.Z. Zhao, R. Mallipeddi, P.N. Suganthan, Multi-objective optimization using self-adaptive differential evolution algorithm, in: IEEE Congress on Evolutionary Computation, 2009, pp. 190–194.
[201] K. Li, Á. Fialho, S. Kwong, Multi-objective differential evolution with adaptive control of parameters and operators, in: C.C. Coello (Ed.), Learning and Intelligent Optimization, vol. 6683, Springer Berlin Heidelberg, 2011, pp. 473–487.
[202] K. Li, S. Kwong, R. Wang, J. Cao, I.J. Rudas, Multi-objective differential evolution with self-navigation, in: IEEE International Conference on Systems, Man, and Cybernetics, 2012, pp. 508–513.
[203] E. Krempser, Á. Fialho, H.C. Barbosa, Adaptive operator selection at the hyper-level, in: Parallel Problem Solving from Nature - PPSN XII, vol. 7492, Springer Berlin Heidelberg, 2012, pp. 378–387.
[204] P.N. Suganthan, Particle swarm optimiser with neighbourhood operator, in: Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406), vol. 3, 1999, pp. 1–1962.