# Hybridizing Differential Evolution and Particle Swarm Optimization to Design Powerful Optimizers: A Review and Taxonomy

Bin Xin, *Member, IEEE*, Jie Chen, *Member, IEEE*, Juan Zhang, Hao Fang, and Zhi-Hong Peng

***Abstract*—Differential evolution (DE) and particle swarm optimization (PSO) are two formidable population-based optimizers (POs) that follow different philosophies and paradigms, which are successfully and widely applied in scientific and engineering research. The hybridization between DE and PSO represents a promising way to create more powerful optimizers, especially for specific problem solving. In the past decade, numerous hybrids of DE and PSO have emerged with diverse design ideas from many researchers. This paper attempts to comprehensively review the existing hybrids based on DE and PSO with the goal of collection of different ideas to build a systematic taxonomy of hybridization strategies. Taking into account five hybridization factors, i.e., the relationship between parent optimizers, hybridization level, operating order (OO), type of information transfer (TIT), and type of transferred information (TTI), we propose several classification mechanisms and a versatile taxonomy to differentiate and analyze various hybridization strategies. A large number of hybrids, which include the hybrids of DE and PSO and several other representative hybrids, are categorized according to the taxonomy. The taxonomy can be utilized not only as a tool to identify different hybridization strategies, but also as a reference to design hybrid optimizers. The tradeoff between exploration and exploitation regarding hybridization design is discussed and highlighted. Based on the taxonomy proposed, this paper also indicates several promising lines of research that are worthy of devotion in future.**

***Index Terms*—Differential evolution (DE), evolutionary optimization, exploration and exploitation, hybridization strategies, memetic algorithms (MAs), particle swarm optimization (PSO), population-based metaheuristics, taxonomy.**

## I. Introduction

OPTIMIZATION has long been a popular topic in scientific research. This is mainly because we human beings are often in pursuit of perfection, such as maximal profit and minimal cost. Besides, the practical problems are vastly varied so that it is hard to find a universal optimizer without any changes to best solve all problems. The diversity of problems contributes to the diversity of optimizers to a large extent. Therefore, it is not unusual to see that numerous optimizers have emerged since Dantzig proposed the simplex method for the linear programming problem in 1947 as an inception of optimization research [1]. Now, increasingly, problems turn out to be nonlinear, nonconvex, multimodal, discontinuous, and even dynamic. For these complicated problems, stochastic optimizers become favored as they have no dependence on the differentiability, continuity, and convexity of objective functions. Besides, many stochastic optimizers have good performance in global optimization. In the family of stochastic optimizers, well-known typical optimizers include the genetic algorithm (GA) [2], evolutionary strategy (ES) [3] (CMA-ES [4] in particular), evolutionary programming (EP) [5], simulated annealing (SA) [6], ant-colony optimization (ACO) [7], particle swarm optimization (PSO) [8], differential evolution (DE) [9], estimation of distribution algorithm (EDA) [10], and so on. All of them have many variants, which have excellent performance. These variants are based on various improvement strategies.

Hybridization is one of the most efficient strategies to improve the performance of many optimizers [11]–[15]. In genetics, hybridization is the process to combine different varieties or species of organisms to create a hybrid (biology). In evolutionary algorithms (EAs), hybridization refers to merging two or more optimization techniques into a single algorithm. In the research field of combinatorial optimization, hybrid optimizers are also termed as hybrid metaheuristics [15]–[18]. In the past decade, hybrid optimizers have attracted persistent attention from scholars that are interested in design of optimizers and their applications [11]–[18]. As Raidl claimed in his unified view of hybrid metaheuristics [18], it seems that choosing an adequate hybrid approach is determinant to achieve top performance in solving most difficult problems.

Like the crossover operator in GAs [2], hybridization contributes to population diversity but the population here means the family of optimizers. There are many successful examples of hybridization in the evolution process of this family (e.g., [19]–[28]). A common template for hybridization is provided by memetic algorithms (MAs), which combine the respective advantages of global search and local search (LS) [19]. Due to excellent performance, MAs have been favored by many scholars in their research on different optimization problems [29]. However, MAs only represent a special class in the family of

Manuscript received March 21, 2011; accepted June 23, 2011. Date of publication August 8, 2011; date of current version August 15, 2012. This work was supported by the National Science Fund for Distinguished Young Scholars under Grant 60925011 and the National Natural Science Foundation of China 60805035/F0306. This paper was recommended by Associate Editor J. Lazansky.

B. Xin is with the School of Automation, Beijing Institute of Technology, and with the Key Laboratory of Complex System Intelligent Control and Decision, Ministry of Education, Beijing 100081, China, and also with the Decision and Cognitive Sciences Research Centre, Manchester Business School, the University of Manchester, Manchester M15 6PB, U.K. (e-mail: brucebin@bit.edu.cn).

J. Chen, J. Zhang, H. Fang, and Z.-H. Peng are with the School of Automation, Beijing Institute of Technology, and also with the Key Laboratory of Complex System Intelligent Control and Decision, Ministry of Education, Beijing 100081, China (e-mail: chenjie@bit.edu.cn; zhjuan@bit.edu.cn; fangh@bit.edu.cn; peng@bit.edu.cn).

Digital Object Identifier 10.1109/TSMCC.2011.2160941

hybrid optimizers. There are manifold possibilities of hybridizing different optimizers, which follow diverse philosophies and paradigms.

This review paper aims to establish a comprehensive taxonomy of hybridization strategies by the review of existing hybrid optimizers that are based on two powerful POs—DE and PSO. To build a unified nomenclature, we term any optimizer that is based on hybridization between DE and PSO as DEPSO. To date, it is impossible to list all kinds of hybrids within a paper of tens of pages, since there are too many hybrids that exist in the literature (about 1450 000 related works according to the Google search result with the index terms "hybrid algorithm" and "optimization"). One reason we choose DEPSO as the representative for hybrid optimizers to build the taxonomy is that both PSO and DE are POs that involve masses of hybridization factors. Besides, a large number of DEPSO variants have been proposed in the literature, laying a solid foundation for the establishment of a comprehensive taxonomy. Moreover, as pointed out later, the proposed taxonomy that is based on DEPSO can be easily applied to other hybrid optimizers. In order to provide a comprehensive reference to design powerful hybrid optimizers, we pool diverse ideas of scholars who have contributed to DEPSO's development. The contribution of this paper is that it provides readers not only a review on hybridization of DE and PSO, but more importantly a broader sight and a tool on how to differentiate and design different hybrid optimizers by a comprehensive systematic taxonomy of hybridization strategies.

This paper is structured as follows. In Section II, a brief introduction on PSO and DE is presented. In Section III, previous works on DEPSO are reviewed. In Section IV, a new taxonomy of hybridization strategies is proposed in an attempt to provide a common terminology and classification mechanisms. Most existing DEPSOs to date and other typical hybrid optimizers are categorized according to the taxonomy. In Section V, we present a discussion about some common issues of hybrid optimizers (not restricted to DEPSO) and indicate a number of topics that are worthy of future research. In Section VI, we conclude the paper.

## II. Differential Evolution and Particle Swarm Optimization

### A. Classical Differential Evolution

The DE proposed by Storn and Price [9], [30] is a formidable PO, which has been widely applied in practice [30]–[35]. It was established on the framework of GAs and inspired by the *Nelder–Mead* simplex method [30]. It has three operators—mutation, crossover, and selection—which are similar to GAs. However, the mutation in DE is a distinct innovation. It is based on the difference of different individuals, borrowing ideas from the Nelder–Mead simplex method. A general notation for DE is DE/$x$/$y$/$z$, where $x$ specifies the base vector to be mutated, $y$ is the number of difference vectors used, and $z$ denotes the crossover scheme [30]. The most classical DE variant is DE/rand/1/bin. In this variant, for the mutation of the $i$th individual in the DE population$\{\mathbf{x}_i|i=1,2,\ldots,\mathrm{PS}\}$, three different individuals $\mathbf{x}_{r1}$, $\mathbf{x}_{r2}$, and $\mathbf{x}_{r3}$ with $r1\neq r2\neq r3\neq i$ will be randomly (rand) chosen from the population to generate a new vector. The new vector can be expressed as follows:

$$\mathbf{z}_i=\underbrace{\mathbf{x}_{r1}}_{\text{base}}+F\cdot\underbrace{(\mathbf{x}_{r2}-\mathbf{x}_{r3})}_{\text{indiv.diff.}} \tag{1}$$

where $F$ is the so-called scaling factor, which is a positive constant. A general setting for this factor is $F\in[0,2]$. However, Storn and Price suggest $F\in[0.5,1]$ as such a setting may result in good optimization effectiveness [9]. In addition, there are many advanced DE variants, which are based on the dynamic tuning of this parameter (e.g., [36] and [37], see also the survey [32] on DE for more introduction).

After mutation, a binominal (bin) crossover operates on the vector $\mathbf{z}_i$ and the target vector $\mathbf{x}_i$ to generate the final vector $\mathbf{u}_i$ in the following way:

$$u_{i,d}=\begin{cases}z_{i,d}, & \text{if rand}_i^d\leq \mathrm{CR}\text{ or }d==rn_i;\\ x_{i,d}, & \text{otherwise}\end{cases} \tag{2}$$

where $x_{i,d}$, $z_{i,d}$, and $u_{i,d}$ are the $d$th dimensional components of the vectors $\mathbf{x}_i$, $\mathbf{z}_i$, and $\mathbf{u}_i$, respectively; CR is the predefined crossover probability, which is usually set to a fixed value in (0,1) or changes dynamically within (0,1); $rn_i$ is a number that is randomly selected from the index set $\{1,2,\ldots,D\}$ and used to ensure that the trial vector $\mathbf{u}_i$ is different from the original solution $\mathbf{x}_i$.

Finally, $\mathbf{u}_i$ will be compared with $\mathbf{x}_i$, and the better one will be selected to be a member of the DE population for the next generation. This replacement scheme is *de facto* a one-to-one tournament selection.

Another typical DE variant is DE/best/2/bin [9]. It shares much similarity with DE/rand/1/bin except for the mutation strategy. The DE/best/2 mutation can be described as follows:

$$\mathbf{z}_i=\underbrace{\mathbf{x}_{\text{best}}}_{\text{base}}+F\cdot\underbrace{(\mathbf{x}_{r1}-\mathbf{x}_{r2}+\mathbf{x}_{r3}-\mathbf{x}_{r4})}_{\text{individual difference}} \tag{3}$$

where $\mathbf{x}_{\text{best}}$ is the best individual in the current population; $\mathbf{x}_{r1}$, $\mathbf{x}_{r2}$, $\mathbf{x}_{r3}$, and $\mathbf{x}_{r4}$ are four different individuals that are randomly chosen from the current population. The following two DE mutation strategies are also used in the literature.

1) DE/current-to-best/1(DE/target-to-best/1) [30], [37]

$$\mathbf{z}_i=\underbrace{\mathbf{x}_i+\lambda\cdot(\mathbf{x}_{\text{best}}-\mathbf{x}_i)}_{\text{base}}+F\cdot\underbrace{(\mathbf{x}_{r1}-\mathbf{x}_{r2})}_{\text{individual difference}}. \tag{4}$$

2) DE/mid-to-better/1 [38], [39]

$$\mathbf{z}_i=\underbrace{(\mathbf{x}_i+\mathbf{x}_{\text{better}})/2+\lambda\cdot(\mathbf{x}_{\text{better}}-\mathbf{x}_i)}_{\text{base}}+F\cdot\underbrace{(\mathbf{x}_{r1}-\mathbf{x}_{r2})}_{\text{individual difference}}. \tag{5}$$

Here, $\mathbf{x}_{\text{better}}$ is an individual that is randomly selected from the DE population, and its fitness is better than (or equal to) that of $\mathbf{x}_i$; and $\lambda$ is a scale coefficient with $0<\lambda<1$, and it is often set to the same value as $F$. More DE variants can be found in [30] and [31].

### B. Classical Particle Swarm Optimization

The particle swarm optimizer that is proposed by Kennedy and Eberhart is also a powerful tool in search and optimization [8]. During the past decade, PSO has been successfully and widely applied in the practice of science and engineering (e.g., [40]–[43]), which demonstrates the superiority of this algorithm. Originally, PSO was inspired by the cooperative behavior of animal groups, such as bird flocks. In PSO, many particles form a swarm, which flies through $D$-dimensional problem space. Each particle has its own velocity and fitness value. Particles accumulate their own experiences about the problem space and, meanwhile, learn from each other according to their fitness values. The iteration equations for the velocity and position in PSO with inertia weight (PSO-w) [44] are given as follows:

$$v_{i,d}^{k+1} = \underbrace{w \cdot v_{i,d}^{k}}_{\text{inertia}} + \underbrace{c1 \cdot \text{rand1} \cdot (\text{pbest}_{i,d}^{k} - x_{i,d}^{k})}_{\text{individual cognition}} + \underbrace{c2 \cdot \text{rand2} \cdot (\text{nbest}_{i,d}^{k} - x_{i,d}^{k})}_{\text{social learning}} \quad (6)$$

$$x_{i,d}^{k+1} = x_{i,d}^{k} + v_{i,d}^{k+1} \quad (7)$$

where $\mathbf{x}_i^k = [x_{i,1}^k, x_{i,2}^k, \ldots, x_{i,D}^k]^T$ and $\mathbf{v}_i^k = [v_{i,1}^k, v_{i,2}^k, \ldots, v_{i,D}^k]^T$ represent the position and velocity of the $i$th particle at the $k$th generation ($i = 1, 2, \ldots, \text{PS}$; PS is the population size), respectively; $\mathbf{pbest}_i^k = [\text{pbest}_{i,1}^k, \text{pbest}_{i,2}^k, \ldots, \text{pbest}_{i,D}^k]^T$ is the best position that is found so far by the $i$th particle; $\mathbf{nbest}_i^k = [\text{nbest}_{i,1}^k, \text{nbest}_{i,2}^k, \ldots, \text{nbest}_{i,D}^k]^T$ is the best position that is found by particles in the neighborhood of the $i$th particle; and rand1 and rand2 are the random numbers that are uniformly distributed in [0,1].

The PSO parameter $w$ is the so-called inertia weight, and a common setting of this parameter is $w^* = 0.7298$, which corresponds to the constriction factor that is proposed by Clerc and Kennedy [45]. Concerning another setting that is frequently adopted in the literature, the inertia weight is dynamically adjusted, decreasing from a number around 1 to a smaller number around zero [46], [47]. $c1$ and $c2$ are acceleration coefficients, which are also termed as the cognitive factor and the social factor, respectively. Both of them are set to $0.7298 \times 2.05 \approx 1.496$ in the PSO with the constriction factor (PSO-cf) [45].

One kernel of PSO is *population topology*, which is also termed as neighborhood or information topology, specifying the social structure of particle swarm [48]–[50]. Information such as the best position that is found by some particle can only be transferred to its neighbors that are determined by population structure. The population topology has obvious effects on the spreading speed of information and the convergence of particle swarm [48]. Different population topologies correspond to different PSO versions [49], [50]. The most well-known topology models are *Gbest* and *Lbest*. In the Gbest model, each particle's neighborhood covers the whole swarm and all particles are connected with each other. Usually, "$\mathbf{gbest}$"is used to represent the best position found by the whole swarm. In the Lbest model, particles are not fully connected and each particle has fewer neighbors than its Gbest counterpart. The "Ring" topology in which each particle is connected to two neighbors to form a "ring," and the *von Neumann* topology in which each particle has four neighbors on a two-dimensional (2-D) lattice (left, right, above, and below) are typical Lbest topologies [50]. Obviously, the Gbest model can be regarded as a special case of the Lbest model.

### C. Unified Views on Particle Swarm Optimization and Differential Evolution

A unified view is advantageous to understand the similarities and differences of diverse optimizers and identify the fundamental ingredients for hybridization. From the introduction given earlier, it is easy to see that both DE and PSO are POs, manipulating multiple solutions in parallel. Sun *et al.* pointed out that the essence of population-based optimizers is to construct the *sampling probability distribution* from which the next population is generated [51]. Regarding the sampling distribution of DE and PSO, several scholars have conducted some pioneering research [52]–[55]. Ali *et al*. derived the probability distribution of trial points in DE and proposed a point generation scheme to approximate the distribution [52]. Poli analyzed the sampling distribution of a classical PSO with the assumption of *stagnation* and used it to explain the search behavior of PSO [53]. Particularly, Kennedy proposed a simplified PSO version [named *barebones* PSO (BBPSO)] with an attempt to more concisely reflect the essence of the PSO paradigm, i.e., the social interaction of individuals [54]. In this compact PSO, the velocity formula is eliminated, while the sampling distribution of classical PSO is retained by the employment of a *Guassian* sampling approach. Kennedy suggests the normal distribution $x_{i,d}^{k+1} \sim N\left((\text{pbest}_{i,d}^k + \text{nbest}_{i,d}^k)/2, |\text{pbest}_{i,d}^k - \text{nbest}_{i,d}^k|\right)$ for the *Guassian* sampling on each dimension [54]. It is evident that the direct sampling approach still encompasses the collaborative interaction of population members [55]. Omran *et al*. borrowed the idea of Kennedy's BBPSO to design a self-adaptive barebones DE based on DE/rand/1/bin [56]. The base vector and individual difference of DE/rand/1/bin are taken as the mean and standard deviation of the Gaussian sampling distribution for the barebones DE, respectively.

In terms of the sampling approach, there is a big difference between DE and PSO. The sampling of PSO integrates *individual cognition* (pbest) and *social collaboration* (nbest). In contrast, individual experience (pbest) is seemingly not utilized in DE to guide its sampling behavior. DE relies on differential mutation and crossover to generate trial points by means of the usage of the difference of randomly selected individuals to perturb base vectors. However, it is worth noting that the one-to-one tournament selection that is employed by DE ensures any individual to be its own up-to-date pbest. Stated another way, at any time $k$, $\mathbf{pbest}_i^k = \mathbf{x}_i^k$ ($i = 1, 2, \ldots, \text{PS}$) hold in DE. (This is usually not the case for PSO.) In view of this fact, we can deem the sampling of DE as a combination of pbest and perturbed base vectors [see (1) and (2)]. Particularly, in DE/best/2/bin, the base vector is the so-far-best individual (i.e., $\mathbf{gbest}$). Therefore, this DE variant shares some similarity with the Gbest PSO in terms of the sampling approach.

In [57], Xie and Zhang proposed a swarm algorithm framework (SWAF) that is realized by agent-based modeling, regarding each individual (agent) as a barebones cognitive architecture, which gains knowledge by the appropriate deployment of a set of simple rules. In fact, the SWAF provides us a template from the perspective of cognition to comprehend the similarities and differences of diverse population-based metaheuristics that include PSO and DE. In the SWAF, PSO, and DE, to employ different generate-and-test rules, are both regarded as socially biased individual learning heuristics [57].

On the implementation of diverse metaheuristics, Taillard *et al*. set forth a unifying view from the perspective of *adaptive memory-based programming* [58]. Although the concept of memory is usually regarded as a kernel in tabu search [59] and it is usually not stressed in the DE and PSO literature, both DE and PSO have to maintain a memory to record personal best information. Calégari *et al*. proposed a taxonomy of EAs, taking into account multiple classification criteria, such as population sizing strategy and population structure [60]. Both DE and PSO can adopt a *static* (constant) or *dynamic* population size [61]–[65]; howbeit, they differ in terms of population structure as DE originally does not differentiate population topologies.

Greistorfer and Vo$\beta$ proposed a "pool" template to cover diverse classes of metaheuristics [18], [66]. The template uses the metaphor of "pool" to describe the set of candidate solutions that are possibly chosen as ingredients for subsequent recombination methods or as start solutions for improvement methods (IMs). In fact, the pool is a concept that is equivalent to the memory, which Taillard *et al*. emphasized in [58]. Various metaheuristics can be differentiated in terms of IM, *solution combination method* (SCM), and the *use and update of the pool*. We can also comprehend the similarities and differences between DE and PSO according to this template. First, as mentioned earlier, neither classical DE nor classical PSO employs IMs though there exist several advanced variants of DE and PSO, which incorporate *LS techniques* as an improvement strategy [67]–[71]. Second, DE takes differential mutation and crossover as its SCM, while the SCM of PSO is a random combination of pbest and nbest.

As for the pool, both DE and PSO keep only a pool of pbest information, which contains nbest and gbest information, that is to say, any visited solution that is inferior to pbest will be discarded and not used in subsequent samplings. The use and update of the pool here implies a somewhat *greedy* selection scheme and a higher selection pressure. Purely relying on the *pbest* information to guide the search may result in loss of valuable information that is contained in other samplings and miss some opportunities to achieve a more efficient exploration or exploitation. An in-depth utilization of the search history that includes those inferior solutions provides us a viable way to improve the performance of optimizers.

To summarize, DE and PSO are similar in terms of the basic features of POs. They differ mainly in terms of sampling approach. DE depends on a difference-based perturbation and crossover to achieve its sampling, while PSO follows the paradigm to combine individual cognition and population-topology-dependent social learning.

## III. Previous DEPSOs

Following the terminology in GAs, we regard DE and PSO as the parents of DEPSO. As a burgeoning optimizer, DEPSO has shown its prominent advantage and prosperity, which are witnessed by the diversity of DEPSO variants and their applications [28], [72]–[126]. Besides, DEPSO has been further hybridized with other optimizers, giving birth to more complicated hybrids [127]–[129]. In the past decade, many scholars have made contributions to DEPSO research. In the sequel, we will review existing DEPSOs by grouping them into three categories according to their basic features: 1) collaboration-based DEPSO; 2) embedding-based DEPSO; and 3) assistance-based DEPSO. The classification criterion here only considers the *relationship* between parent optimizers for hybridization.

*Collaboration* here means that parent optimizers cooperate with each other in *problem space* to seek the optimum, they *share* or *exchange* accumulated information during their search, and their *own* operating manners are *maintained*, respectively. In contrast, the operating manners of parent optimizers in embedding-based hybrids will be *changed* and cannot be separated *explicitly*. In this case, it is hard to separate their contribution to fitness improvement since the parents are *integrated* into a *holistic* hybrid optimizer. In contrast, *assistance* leads to a very special hybridization strategy in which one parent will not generate new sampling points in search space but serve as an assistance of another parent. For example, one parent under the assistance relationship may work in the *parameter space* of another one. It should be noted that a point in search space can be regarded as a sampling point only if it undergoes fitness evaluation. Usually, the assistant will keep its *own* operating manner, which is a very important feature for distinguishing "assistance" from "embedding." The following is a mathematical formulation of the three kinds of parent relationship:

$$\text{Collaboration}\begin{cases}\text{Parent } A & f_A : S_s^{(n_A(t))} \mapsto S_s^{(n_A(t+1))}[S_{\text{ob}}] \\ \text{Parent } B & f_B : S_s^{(n_B(t))} \mapsto S_s^{(n_B(t+1))}[S_{\text{ob}}]\end{cases}$$

Assistance-Parent B

$$f_B : S_s^{(n_A(t))} \xrightarrow{\text{Parent}A\, f_A : S \mapsto S} S_s^{(n_B(t+1))}[S_{\text{ob}}]$$

Embedding-Parent A and B

$$f_{A\oplus B} : S_s^{(n_A(t))} \mapsto S_s^{(n_B(t+1))}[S_{\text{ob}}]$$

where $f_A$ and $f_B$ represent the mapping of parents A and B, respectively; $S_s$ and $S_{\text{ob}}$ are the solution space and the objective space, respectively; $S_s^{(n)} = \underset{1}{S_s} \times \underset{2}{S_s} \times \cdots \times \underset{n}{S_s}$; $n(t)$is the population size (the optimizer adopts a static population sizing strategy if $n(t) = \text{const}$, and a dynamic strategy otherwise); $[S_{\text{ob}}]$ indicates that solutions that are generated will go through objective evaluation. In the case of assistance, parent **A** as the assistant of parent **B** can work in different spaces, i.e., $S \in \{S_p, S_{op}, S_s, S_{it}, \ldots\}$ (where $S_p$ is parameter space, $S_{\text{op}}$ is operator (optimizer) space, and $S_{\text{it}}$ is information topology space), and the mapping $f_A$ reflects the *complete* operation of parent **A**. The whole mapping in the case of assistance can be regarded as a *composite* mapping $f_B \circ f_A$, and the operations

of parents can be differentiated. In the case of embedding, $f_{A\oplus B}$ represents a *nonseparable* hybrid mapping in which parents A and B are *integrated* into a new *holistic* operator.

In order to differentiate various DEPSOs conveniently, we will name a given DEPSO by combining the initial of the last name of each scholar, who created the DEPSO, into a suffix that follows the terminology "DEPSO."

### A. Collaboration-Based DEPSOs

In [72], a combined swarm DE algorithm (namely DEPSO-H) is proposed, which is a hybrid optimizer based on PSO and DE. To the best of our knowledge, DEPSO-H is the first DEPSO that is reported in the literature. In this DEPSO, particle positions are updated only if their offspring have better fitness. DE acts on the current positions of particles in the PSO swarm at specified intervals. Here, the DE algorithm is DE/rand/1/bin, and the PSO algorithm is a variant of the Gbest PSO. A very similar DEPSO (namely DEPSO-RJAMB) was proposed by Ramesh *et al*. to solve economic dispatch problems [73]. Its DE and PSO parents share and update the current positions of particles in their *respective* manners. Ramesh *et al*. claimed that DEPSO-RJAMB combines the *vibrancy* and *explorative* nature of PSO with the superior *exploitative* nature of hybrid DE (HDE). Note that the HDE is *de facto* a *memetic* DE that is proposed by Lin *et al*., which hybridizes DE/rand/1/bin with a LS method (steepest descent search) [130].

Another typical example is the DEPSO that is proposed by Zhang and Xie [28] (namely DEPSO-ZX). In this DEPSO, PSO and DE also alternate in a *deterministic* way, but DE acts on the *personal best* positions of particles in contrast to the *current* particle positions in DEPSO-H. DE follows PSO at each generation, and it is expected to further improve the personal best positions of PSO. Note that the DE and PSO in DEPSO-ZX are DE/best/2/bin and the classical PSO with the Gbest model, respectively. The primary operations of DEPSO-ZX can be described, in sequence, by (6) and (7) and the following DE operations:

$$\mathbf{z}_i = \mathbf{x}_{\text{best}} + F\cdot(\mathbf{pbest}_{r1} - \mathbf{pbest}_{r2} + \mathbf{pbest}_{r3} - \mathbf{pbest}_{r4}) \tag{8}$$

$$u_{i,d} = \begin{cases} z_{i,d}, & \text{if } \text{rand}_i^d \le \text{CR or } d == rn_i \\ \text{pbest}_{i,d}, & \text{otherwise} \end{cases} \tag{9}$$

where the denotations are the same as those in (2) and (3).

DEPSO-ZX has been successfully applied in multimodal image registration [74], modeling of gene regulatory networks [75], structure optimization of a high-temperature superconducting cable [76], and design of finite-impulse response filters [77]. Besides, DEPSO-ZX was even further hybridized with PSO and ES to form a more complicated hybrid [128], [129]. In [78], Moore and Venayamoorthy proposed a DEPSO (namely DEPSO-MV), which shares the same idea with DEPSO-ZX, but its DE and PSO parents are DE/rand/2/bin and a modified PSO with "Ring" topology, respectively. DEPSO-MV is used in the multiobjective optimization of combinational logic circuits design.

In [79], a DEPSO (namely DEPSO-LWJH) that is similar to DEPSO-ZX is proposed and it is used to train artificial neural networks. Like DEPSO-ZX, the PSO in this hybrid is also based on the Gbest model. Differently, its DE parent is DE/current-to-best/1/bin. In particular, DEPSO-LWJH adopts a chaotic LS to improve its local exploitation ability. Huang *et al*. also adopted the same hybridization strategy to design their DEPSO (namely DEPSO-HWLR) [80]. Besides, DEPSO-HWLR incorporates the technique of the parallel finite-element method (PFEM) to address the back-analysis issue of mechanics parameters in engineering projects. Other analogs of DEPSO-ZX include the DEPSO-LCW proposed by Liu *et al*. to handle constrained numerical and engineering optimization problems [81], and the DEPSO-XXW proposed by Xu *et al*. for partitional clustering [82]. In [83], a similar scheme of evolution of the personal experience of particles (pbest) by DE is proposed (the resultant DEPSOs are all named DEPSO-EPV); however, DE will only be applied to the personal best positions, which have been improved as compared with previous generation.

Liu *et al*. used DE to perturb the positions of particles with the purpose of keeping population diversity [84]. In their DEPSO (namely DEPSO-LHTC), DE follows PSO in each generation to change the current positions of particles. Xu and Gu also proposed a collaboration-based DEPSO, namely DEPSO-XG, in which PSO and DE evolve the whole population alternately [85]. In particular, Xu and Gu incorporated the *average* $\mathbf{pbest}$ position and velocity of particles into the velocity regulation of the classical PSO [cf., (10)]. The DE operations in DEPSO-XG involve two populations: the *original* DE population and an *extra* population to record those trial vectors *inferior* to their target vectors. The "inferior" vectors will be reused in a so-called prior crossover operation to generate intermediate vectors and update particle positions:

$$\begin{aligned} v_{i,d}^{k+1} = {} & \underbrace{w\cdot v_{i,d}^k}_{\text{inertia}} + \underbrace{c1\cdot\text{rand1}\cdot(\text{pbest}_{i,d}^k - x_{i,d}^k)}_{\text{individual cognition}} \\ & + \underbrace{c2\cdot\text{rand2}\cdot(\text{nbest}_{i,d}^k - x_{i,d}^k)}_{\text{social learning}} \\ & + \underbrace{c3\cdot\text{rand3}\cdot(\overline{\text{pbest}_d^k} - x_{i,d}^k)}_{\text{average pbest attraction}} \\ & + \underbrace{c4\cdot\text{rand4}\cdot(\overline{v_d^k} - v_{i,d}^k)}_{\text{average velocity attraction}} \end{aligned} \tag{10}$$

where $\overline{\text{pbest}_d^k} = \frac{\sum_{i=1}^{\text{PS}}\text{pbest}_{i,d}^k}{\text{PS}}$ and $\overline{v_d^k} = \frac{\sum_{i=1}^{\text{PS}} v_{i,d}^k}{\text{PS}}$.

Ning *et al*. introduced differential mutation operator into a basic PSO in order to alleviate the so-called premature convergence of the classical PSO [86]. At each generation of their DEPSO (namely DEPSO-NZL) [86], PSO runs first, and DE/best/2/bin follows to improve the current positions of particles.

Caponio *et al*. designed a DEPSO (namely DEPSO-CNT) by the use of PSO to provide DE a so-called super-fit individual [87]. DEPSO-CNT was established on a DE framework that is hybridized with PSO and two local searchers. PSO works only

at the beginning of the whole search process, refining a part of the initial population and replacing the worst solution with the refined best one. Local searchers will be adaptively chosen by means of a probabilistic scheme, which relies on an index that measures the quality of the super-fit individual with respect to the rest of the population [87]. The performance of DEPSO-CNT was validated in the optimal drive design for a direct current motor and the design of a digital filter. Ali *et al*. also proposed a *two-phase* DEPSO (namely DEPSO-APA) in which DE mainly in charge of *exploration* runs first [88]. DEPSO-APA will switch from DE/rand/1/bin to basic PSO (Gbest topology) according to a predefined threshold for the identification of DE's *convergence*. Once the fitness difference of the best and worst DE individuals reaches the threshold, PSO will act on half of the DE population that consists of elite solutions to achieve the final exploitation of promising regions.

Gong *et al*. hybridized DE/best/2/bin with a PSO variant, which removes traditional velocity terms and adopts time-variant acceleration coefficients [89]. Their DEPSO (namely DEPSO-GZQ) was proposed to solve *multiobjective* environmental/economic power dispatch problems. The motivation behind DEPSO-GZQ is an effective synergy of PSO's *exploration* ability and the *exploitation* ability of DE/best/2/bin. The DE and PSO parents of DEPSO-GZQ share and update an *external archive,* which records *nondominated* solutions that are found so far. Particularly, the *crowding distance*, as an index of the *solution diversity* in multiobjective space, is utilized to differentiate the quality of nondominated solutions. Grobler and Engelbrecht also proposed a DEPSO, namely DEPSO-GE, to solve multiobjective optimization problems [90]. Like DEPSO-GZQ, the DE and PSO parents of DEPSO-GE also interact via an external archive. However, they are assigned different objectives to optimize. Another DEPSO for multiobjective optimization is the DEPSO-JC that is proposed by Jiang and Cai [91], employing PSO and DE as regeneration methods. A new acceptance rule that is called *distance/volume fitness* was proposed to update the external archive of the multiobjective DEPSO.

Niu and Li proposed a coevolutionary DEPSO, namely DEPSO-NL, employing DE and PSO to evolve their respective subpopulations. Its DE parent (DE/rand/1/bin) and PSO parent (classical Gbest PSO) interact by sharing the global best solution that is found so far [92], [93]. The same idea was also adopted by Zhang *et al*. who created a chaotic coevolutionary DEPSO (namely DEPSO-ZZS) [94]. In particular, DEPSO-ZZS employs Tent-map-based chaotic perturbation, which follows both DE (DE/current-to-best/1/bin) and PSO (Gbest) to update DE individuals and particle positions, respectively. Wang *et al*. also proposed a similar DEPSO (namely DEPSO-WYZ), which maintains three subpopulations that are evolved, respectively, by classical PSO and two DE variants [95]. The three subpopulations share the global best solution during their evolution. Another analog is the DEPSO-SC that is proposed by Sentinella and Casalino, incorporating three EAs, which include the GA, PSO, and DE, to solve spacecraft trajectory optimization problems [96], [97]. The three optimizers work in parallel and, periodically, let their best individuals migrate to other subpopulations. Yang *et al*. hybridized a quantum-behaved PSO with DE/rand/2/bin to create a coevolutionary DEPSO (namely DEPSO-YMN) in which the DE parent evolves elite individuals and the PSO evolves common individuals [98]. The evolution of common individuals is guided by the pbest of elite solutions, while the DE-based evolution of elites contains the perturbation based on the difference of common individuals. Recently, Peng *et al*. proposed a general coevolutionary framework, namely population-based algorithm portfolio (PAP), to combine different population-based search algorithms [99]. Four optimizers, which involve a self-adaptive DE and classical PSO, are chosen as candidate parent optimizers to test the efficiency of PAP. Within the PAP framework, parents exchange the elite solutions that are distributed in different subpopulations by the migration mechanism usually employed in niching GAs. Their findings based on experimental analysis demonstrate that the PAP has obvious superiority over the homogeneous parallel hybridization of any parent optimizer. Peng *et al*. also pointed out that the complementarity of parents is a key issue to ensure the superior performance of their PAP [99].

In [100], a novel coevolutionary DEPSO (namely DEPSO-HW) is proposed based on diploid genetic theory to solve the open vehicle routing problem. In DEPSO-HW, PSO evolves the so-called dominant chromosomes, while DE evolves recessive chromosomes. When a recessive chromosome is better than its corresponding dominant chromosome, they will be exchanged into opposite subpopulations. Wu *et al*. proposed a distinctive coevolutionary DEPSO (namely DEPSO-WGHZ) by introducing the mutation operator of DE within the framework of the cultural algorithm into PSO [101]. The average fitness of all particles is utilized to divide the whole swarm into two subpopulations, which are evolved, respectively, by PSO and DE [101]. The DE parent of DEPSO-WGHZ resembles DE/rand/1/bin, but its crossover operator is removed. In addition, the differential mutation that is affected by three kinds of knowledge in belief space is implemented as an influence function to evolve inferior particles whose fitness is lower than average.

Recently, Hao *et al*. constructed a new DEPSO (namely DEPSO-HGH) [39]. In this hybrid optimizer, DE and PSO are regarded as two operators to generate candidate solutions, and they act on the level of dimensional components of individuals. Stated another way, each dimensional component of a candidate solution can be generated in either DE or PSO manner. The choice of generation manners depends on a predefined probability (CR) as shown in (11)–(13). In addition, the DE and PSO parents of DEPSO-HGH are DE/mid-to-better/1/bin and the PSO-cf mentioned in Section II:

$$\text{PSO}: \begin{cases} v_{i,d}^{k+1} = w \cdot v_{i,d}^{k} + c1 \cdot \text{rand1} \cdot (\text{pbest}_{i,d}^{k} - x_{i,d}^{k}) \\ \qquad\qquad + c2 \cdot \text{rand2} \cdot (\text{gbest}_{d}^{k} - x_{i,d}^{k}) \\ z_{i,d}^{k+1} = x_{i,d}^{k} + v_{i,d}^{k+1} \end{cases} \tag{11}$$

$$\text{DE}: y_{i,d}^{k+1} = \frac{x_{i,d}^{k} + x_{r1,d}^{k}}{2} + F \cdot (x_{r1,d}^{k} - x_{i,d}^{k} + x_{r2,d}^{k} - x_{r3,d}^{k}) \tag{12}$$

$$u_{i,d}^{k+1} = \begin{cases} y_{i,d}^{k+1}, & \text{if rand} < \text{CR} \\ z_{i,d}^{k+1}, & \text{otherwise.} \end{cases} \tag{13}$$

where the denotations are the same as those in (1)–(7), $\text{gbest}_d^k$ is the $d$th component of the global best position $\mathbf{gbest}$ at the $k$th generation, and it is required that the fitness of $\mathbf{x}_{r1}$ is not worse than that of $\mathbf{x}_i$.

The DEPSO that is designed by De *et al*. (namely DEPSO-DRKC) [102] combined the self-adaptive Pareto DE that is proposed by Abbass *et al*. [131] and a basic PSO with random inertia weight. In the evolutionary loop of DEPSO-DRKC, DE operates on some selected dimensions to refine current particle positions, as well as modulate particle velocities, and PSO follows to regulate the current positions of particles. DEPSO-DRKC was employed to maximize camera coverage area in the coordination of robot ants.

In [103], two DEPSOs are presented. The first DEPSO (namely DEPSO-OES1) is somewhat similar to DEPSO-HGH. The DE (DE/rand/1/bin) and PSO (PSO-cf) in it also alternate in a *stochastic* way, but both DE and PSO act on the level of a whole individual, that is to say, each individual at each generation has only one updating method of its current position (DE or PSO). Besides, the probability of selecting an updating method and the scaling factor in DE are dynamic and adaptive. A similar DEPSO (DEPSO-KL) was proposed by Kim and Lee [104]. At each generation, each individual selects the updating method of its *current* position, either PSO-cf or DE/best/1/bin, according to a predefined probability. Kim and Lee also employed the reinforcement learning method, i.e., Q-Learning, to adjust three primary parameters of DEPSO-KL [104]. Another DEPSO akin to DEPSO-OES1 is the one that is proposed by Dhahri *et al*. (namely DEPSO-DAK) [105]. Its PSO parent will be selected according to a predefined probability to improve the *worst* individuals in current population. DEPSO-DAK was applied to the design of beta basis function neural networks. In the DEPSO that is proposed by Pant *et al*. (namely DEPSO-PTGA) [106], PSO will be activated to evolve an individual only if DE does not bring the improvement of its fitness.

Khamsawang *et al*. proposed four versions of DEPSO (DEPSO-KWJ) with the same goal of the use of DE to improve the exploration ability of PSO [107]. It was claimed that the mutation operators of DE will be *activated* if the velocity values of particles are near zero or violate predefined bounds. According to the formulas that are presented in [107], Khamsawang *et al*. used the difference of individuals, instead of an entire differential mutation, to regulate the velocity of particles. In other words, the difference-based velocity regulation does not involve base vectors. However, since the velocity that is modulated by DE will be added to the current position as shown in (7), we can also regard the operation as a differential mutation with the current position being its base vector. Takano and Hagiwara proposed an *integrated framework* to effectively combine the merits of several evolutionary computations that include the GA, PSO, and DE [108]. The framework holds several individual pools, and each pool corresponds to one EC. Each incorporated EC has its own evaluated value (EV), and it changes according to the best fitness value at each generation. The number of individuals in each EC changes according to the EV. All individuals have their own lifetime to reselect EC, and the probability of each EC to be selected depends on the EV. We term their hybrid, here, as DEPSO-TH for unification though the term GADEPSO seems more suitable to express its hybridization. Xin *et al*. proposed an adaptive DEPSO, namely DEPSO-XCPP, which achieves on-the-fly adaptation of evolution methods for individuals in a statistical learning way, according to the relative success ratio of PSO versus DE in a previous learning period [109]. The idea to employ statistical learning to choose evolution methods can also be applied to the adaptation of control parameters, evolutionary operators (e.g., the differential mutation [37], [132]), and LS techniques [19], [24], [133], [134].

### B. Embedding-Based DEPSOs

A typical embedding-based DEPSO is the PSO with differentially perturbed velocity (namely DEPSO-DKC; PSODV in [110]) that is proposed by Das *et al*. The embedding way in PSODV is shown as follows:

$$v_{i,d}^{k+1} = \begin{cases} w \cdot v_{i,d}^k + \beta \cdot \delta_d + \text{c2} \cdot \text{rand2} \cdot (\text{gbest}_d^k - x_{i,d}^k), \\ \qquad\qquad\qquad\qquad \text{if rand} < \text{CR} \\ v_{i,d}^k, \qquad\qquad\qquad\qquad \text{otherwise} \end{cases} \tag{14}$$

$$u_{i,d}^{k+1} = x_{i,d}^k + v_{i,d}^{k+1} \tag{15}$$

where $\delta_d = x_{r1,d}^k - x_{r2,d}^k$ is the $d$th component of the difference vector of two randomly selected individuals $\mathbf{x}_{r1}$ and $\mathbf{x}_{r2}$ with $r1 \neq r2 \neq i$, $\beta$ is the scaling factor, CR is the predefined crossover probability, and the definitions of $v_{i,d}^k$, $w$, $x_{i,d}^k$, $c2$, rand2, and $u_i^d$ are the same as those mentioned in (1)–(7). Note that the selection operator that is used in DE/rand/1/bin is also employed in this DEPSO to operate on the trial vector $\mathbf{u}_i$ and the corresponding individual $\mathbf{x}_i$. It was reported that DEPSO-DKC has been successfully applied to spatial clustering with obstacles constraints [111] and fault diagnosis of gear-box [112].

Li claimed that the parameter-free HDE that he proposed in [113] (namely DEPSO-L) is derived from the mechanisms of PSO (viz., topologies, inertia weight, neighborhood best, and personal best). The crossover operation of DE is removed, while its selection operation is retained. The primary operations of DEPSO-L are presented in (16) and (17)

$$x_{i,d}^{k+1} = \text{nbest}_{i,d}^k + \text{rand1} \cdot (\text{pbest}_{i,d}^k - x_{i,d}^k) + \text{rand2} \cdot v_{i,d}^k \tag{16}$$

$$\mathbf{v}_i^{k+1} = \begin{cases} \mathbf{v}_i^k, & \text{if } \mathbf{x}_i^{k+1} == \mathbf{x}_i^k \\ \mathbf{x}_i^{k+1} - \mathbf{x}_i^k, & \text{otherwise.} \end{cases} \tag{17}$$

From (16), it seems that DEPSO-L follows the paradigm of differential mutation with nbest being its base vector. However, the difference term in (16) is actually the PSO's individual cognition as shown in (6), not the individual difference that usually exists in diverse DE variants [cf., (1), (3)–(5)]. Besides, the term "velocity" in DEPSO-L turns out to be *passive* since it is derived from the difference of the same particle's positions at

contiguous time instants [cf., (17)], instead of being modulated by inertia, individual cognition, and social learning [cf., (6)].

Shu and Li introduced PSO's *topology* and *velocity regulation* into DE/best/1/bin to construct an embedding-based DEPSO (namely DEPSO-SL) [114]. The primary operations of DEPSO-SL are shown in (18)–(20), with the base vector of DE/best/1/bin (i.e., $\mathbf{gbest}$) being replaced by $\mathbf{nbest}_i$, which is determined by a *Ring* topology. Velocity, here, is defined as the difference of the same trial vector's positions at contiguous time instants [cf., (20)]. From (19), it seems as if there is an inertia part in DEPSO-SL like that in (6); howbeit, the "inertia" here is not used for velocity regulation but for the perturbation of trial vectors:

$$u_{i,d}^{k+1} = \begin{cases} \text{nbest}_{i,d}^{k} + F \cdot (x_{r1,d}^{k} - x_{r2,d}^{k}), \\ \qquad\qquad \text{if rand} < \text{CR or } d == rn_i \\ x_{i,d}^{k}, \qquad\quad \text{otherwise} \end{cases} \tag{18}$$

$$\mathbf{u}_i^{k+1} = \mathbf{u}_i^{k+1} + w \cdot \text{rand} \cdot \mathbf{v}_i^{k} \tag{19}$$

$$\mathbf{v}_i^{k+1} = \mathbf{u}_i^{k+1} - \mathbf{u}_i^{k}. \tag{20}$$

Das *et al*. proposed a neighborhood-based differential mutation operator, which introduces PSO's population topology into DE (the DEPSO variant, here, is termed as DEPSO-DACK) [115]. The primary operations of DEPSO-DACK (i.e., its novel differential mutation operator) are shown as follows:

$$\mathbf{L}_i^{k+1} = \mathbf{x}_i^{k} + \alpha \cdot (\mathbf{nbest}_i^{k} - \mathbf{x}_i^{k}) + \beta \cdot (\mathbf{x}_p^{k} - \mathbf{x}_q^{k}) \tag{21}$$

$$\mathbf{G}_i^{k+1} = \mathbf{x}_i^{k} + \alpha \cdot (\mathbf{gbest}^{k} - \mathbf{x}_i^{k}) + \beta \cdot (\mathbf{x}_{r1}^{k} - \mathbf{x}_{r2}^{k}) \tag{22}$$

$$\mathbf{z}_i^{k+1} = w \cdot \mathbf{G}_i^{k+1} + (1-w) \cdot \mathbf{L}_i^{k+1} \tag{23}$$

where $p, q \in [i-r, i+r]$ with $p \neq q \neq i$ (where $r$ is neighborhood radius), indicating that the individuals for the difference operation in (21) are selected from the neighborhood of the target individual $\mathbf{x}_i^k$. The local-neighborhood-based mutant vector $\mathbf{L}_i^{k+1}$ and the global-neighborhood-based mutant vector $\mathbf{G}_i^{k+1}$ favor exploration and exploitation, respectively. As (23) shows, the hybrid mutant vector $\mathbf{z}_i^{k+1}$ is a balance between exploration and exploitation capabilities.

Xu *et al*. also proposed an embedding-based DEPSO (namely DEPSO-XLFWW) whose primary operation is shown in (24) [116]. DEPSO-XLFWW seems very concise, and it inherits the crossover operation and the format of differential mutation in DE. Strictly speaking, however, it does not employ differential mutation. Instead, it applies a PSO version without velocity and individual cognition, which formally shares certain similarity with the differential mutation operator DE/rand/1:

$$x_{i,d}^{k+1} = \begin{cases} x_{i,d}^{k} + c2 \cdot \text{rand2} \cdot (\text{gbest}_d^{k} - x_{i,d}^{k}), \\ \qquad\qquad \text{if rand} < \text{CR or } d == rn_i \\ x_{i,d}^{k}, \qquad\quad \text{otherwise} \end{cases} \tag{24}$$

where the denotations are the same as those in (1)–(7).

### C. Assistance-Based DEPSOs

Kannan *et al*. [117] proposed a distinctive DEPSO (namely DEPSO-KSSP; C-PSO in [117]). Its PSO parent takes the form of classical PSO [cf., (6) and (7)]. The DE algorithm in it is employed to select three control parameters (i.e., $w$, $c1$, and $c2$) online for PSO. In other words, DE serves as a metaoptimizer for the optimization of PSO's search behavior. The DE population is randomly initialized and, then, evolves along with particle swarm. The evaluation of a DE individual (i.e., a PSO parameter setting) is attached to the fitness evaluation of the corresponding particle. Denote by $\mathbf{para}_i = [w_i, c1_i, c2_i]^T$ the $i$th DE individual. Then, the primary operations of DE in DEPSO-KSSP can be described as follows.

Differential mutation:

$$\mathbf{z}_i^{k} = \mathbf{para}_{r1}^{k} + F \cdot (\mathbf{para}_{r2}^{k} - \mathbf{para}_{r3}^{k}). \tag{25}$$

Crossover:

$$u_{i,d}^{k+1} = \begin{cases} z_{i,d}^{k}, & \text{if rand}_i^d \leq \text{CR or } d == rn_i \\ \text{para}_{i,d}^{k}, & \text{otherwise.} \end{cases} \tag{26}$$

The idea to employ a metaoptimizer to optimize (regulate) the search behavior of another optimizer used to be adopted by Grefenstette for the optimization of GA's control parameters [135]. In Grefenstette's research, a meta-GA is employed to tune the parameters of another GA.

Jiang *et al*. put forward a novel hybridization strategy, which uses DE to operate on the velocity of particles [118]. As Jiang *et al*. pointed out, the population diversity of PSO is highly dependent on the velocity of particles, indicating that an appropriate control scheme of the particle velocity can benefit the global convergence of PSO. The primary operations of their DEPSO (namely DEPSO-JWJ) include basic PSO operations [cf., (6) and (7)] and the following:

$$v_{i,d}^{k+1} = \begin{cases} v_{i,d}^{k} + F \cdot (v_{r1,d}^{k} - v_{r2,d}^{k} + v_{r3,d}^{k} - v_{r4,d}^{k}), \\ \qquad\qquad \text{if rand} < \text{CR or } d == rn_i \\ v_{i,d}^{k}, \qquad\quad \text{otherwise} \end{cases} \tag{27}$$

$$x_{i,d}^{k+1} = \text{rand} \cdot \text{pbest}_{i,d}^{k} + (1 - \text{rand}) \cdot \text{gbest}_d^{k} + v_{i,d}^{k+1}. \tag{28}$$

According to (27), we can term the DE parent of DEPSO-JWJ as DE/current/2/bin by following the nomenclature that is mentioned in Section II-A. The operations that are shown in (27) and (28) will be implemented only if PSO is trapped into stagnation. The operation shown in (28) is not a part of DE but a BBPSO perturbed by the velocity that is generated by DE. To sum up, the DE parent in this DEPSO serves as an assistant of PSO to regulate the velocity of particles.

The second DEPSO that is proposed in [103] by Omran *et al*. (namely DEPSO-OES2) combined the BBPSO that is proposed by Kennedy [54] and DE as follows:

$$x_{i,d}^{k+1} = \begin{cases} p_{i,d}^{k} + \text{rand1} \cdot (x_{r1,d}^{k} - x_{r2,d}^{k}), & \text{if rand} > p_r \\ \text{pbest}_{r3,d}^{k}, & \text{otherwise} \end{cases} \tag{29}$$

$$p_{i,d}^{k} = \text{rand2} \cdot \text{pbest}_{i,d}^{k} + (1 - \text{rand2}) \cdot \text{nbest}_{i,d}^{k} \tag{30}$$

where $p_r$ is the predefined recombination probability, and the other denotations are the same as those in (1)–(7). If the neighborhood topology is based on the Gbest model, the term $\text{nbest}_{i,d}^{k}$ in (30) will be replaced by $\text{gbest}_d^{k}$. The operation of BBPSO is reflected by (30), a direct stochastic combination of $\mathbf{pbest}_i$

and $\mathbf{nbest}_i$ on each dimension. Obviously, the point that is generated by BBPSO, i.e., $\mathbf{p}_i^k = [p_{i,1}^k, p_{i,2}^k, \ldots, p_{i,D}^k]^T$, is taken as the base vector for the DE operation that is reflected by (29). Therefore, the role of BBPSO in DEPSO-OES2 is an *assistant*, providing *base vectors* for DE. DEPSO-OES2 (named barebones DE by its inventors) was applied to unsupervised image classification [119].

Wickramasinghe and Li proposed an assistance-based DEPSO (namely DEPSO-WL) for multiobjective optimization by using DE/rand/1/bin to choose the leaders (i.e., $\mathbf{nbest}_i$) for PSO-cf at each generation [120]. The trial vector that is generated by DE is taken as the leader in PSO. It should be noted that the trial vector will not undergo fitness evaluation, indicating that the selection operator of DE is removed and the trial vector is not a real sampling point. In this sense, DE is an assistant of PSO to provide the ingredient (i.e., $\mathbf{nbest}_i$) for its operation. However, once the trial vector is evaluated, the DE parent will contribute to fitness improvement as a major participant in the search of high-quality solutions rather than merely an assistant of PSO, and in this case, the resultant DEPSO will be based on collaboration.

Some intricate DEPSOs may involve more than one kind of parent relationship. For example, in the DEPSO that is proposed by Zhang *et al*. [121] (namely DEPSO-ZNLOD), DE and PSO on one hand are hybridized in an embedding way and, on the other hand, work together under a collaboration relationship. The DE operations in DEPSO-ZNLOD are shown as follows:

$$\mathbf{z}_i^k = \mathbf{nbest}_i^k + F_i \cdot (\mathbf{pbest}_{r1}^k - \mathbf{pbest}_{r2}^k) \tag{31}$$

$$y_{i,d}^k = \begin{cases} z_{i,d}^k, & \text{if rand} < \text{CR or } d == rn_i \\ x_{i,d}^k, & \text{otherwise} \end{cases} \tag{32}$$

$$\mathbf{x}_i^{k+1} = \mathbf{y}_i^k \tag{33}$$

$$\mathbf{v}_i^{k+1} = \mathbf{x}_i^{k+1} - \mathbf{x}_i^k \tag{34}$$

$$\mathbf{pbest}_i^{k+1} = \begin{cases} \mathbf{x}_i^{k+1}, & \text{if fitness}(\mathbf{x}_i^{k+1}) > \text{fitness}(\mathbf{pbest}_i^k) \\ \mathbf{pbest}_i^k, & \text{otherwise.} \end{cases} \tag{35}$$

It is obvious that PSO's population topology is embedded into DE by taking $\mathbf{nbest}$ as the base vector for differential mutation [cf., (31)]. Therefore, the aforementioned DE is actually an embedding-based DEPSO (cf., DEPSO-DACK [115]). Moreover, DEPSO-ZNLOD employs the classical PSO as a method to evolve the whole population. In fact, the DE works only at specified intervals to fine tune the personal best particle positions [121], and most of the time, the PSO dominates the evolutionary process. Accordingly, the DE and PSO also hold a collaboration relationship.

Liu *et al*. incorporated DE/rand/1/bin and PSO into the framework of cultural algorithm [122], and the resultant hybrid is named DEPSO-LWY here. A primary operation is shown in (36), which reflects the hybridization of DE and PSO based on both *embedding* (DE's mutation is perturbed by PSO's velocity) and *collaboration* (PSO and perturbed DE are further hybridized on component level). Besides, unlike most PSO variants, the particle positions and velocities are modulated according to four kinds of knowledge sources that are contained in the belief space of the cultural algorithm. Particle positions will be drawn toward the global best position (i.e., $\mathbf{gbest}$), and particle velocities will be confined by lower and upper bounds that are dependent on personal best particle positions. In this sense, the PSO is further hybridized with the DEPSO that is shown in (36) under a collaboration relationship:

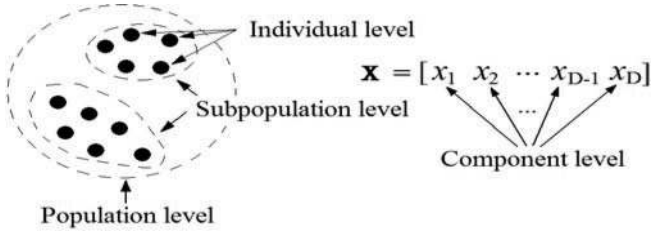



Fig. 1. *Hybridization levels*: Black spots represent contemporary solutions that are distributed in search space. $\mathbf{X}$ represents a solution in $D$-dimensional space and $x_i$ $(i = 1, 2, \ldots, D)$ are the components of $\mathbf{X}$.

$$x_{i,d}^{k+1} = \begin{cases} x_{r1,d}^k + F \cdot (x_{r2,d}^k - x_{r3,d}^k) + v_{i,d}^{k+1}, & \text{(perturbed DE)} \\ \qquad\qquad \text{if rand} < \text{CR or } d = rn_i \\ x_{i,d}^k + v_{i,d}^{k+1}, & \text{(PSO)otherwise.} \end{cases} \tag{36}$$

Although the aforementioned multifarious DEPSOs were constructed with the same purpose to combine two or more excellent optimizers to create a more advanced one, they have more or less differences in their hybridization strategies and parent optimizers. A systematic taxonomy, as shown in the next section, will be beneficial to analyze different kinds of DEPSOs in a comprehensive way.

## IV. Taxonomy on Hybridization Strategies

A favorable taxonomy of hybridization strategies should be capable of differentiating different strategies as well as providing designers a convenient and efficient means of determining a hybridization scheme. In this section, we will set forth the elements of a hybridization strategy (i.e., hybridization factors) and give a classification of existing DEPSOs and some other prevailing hybrids.

### A. Hybridization Factors

In Section III, we have revealed a basic classification criterion—the relationship between parent optimizers. Here, another four hybridization factors are taken into account: the hybridization level (HL) of parent optimizers, the OO of parent optimizers, the TIT, and the TTI.

*1) Hybridization Level:* The HL of parent optimizers, which are highly dependent on their respective operating levels (OLs), refers to the OL at which they are hybridized, indicating the *spatial* assignment of parent optimizers. As illustrated in Fig. 1, there are four kinds of HLs—*component* level, *individual* level, *subpopulation* level, and *population* level (in order from the lowest to the highest level). This classification is also applicable to the OL. The OL of a parent optimizer is determined by the

object, which can, according to hybridization design, apply the optimizer without being required to use the same optimizer with any other objects. If the object is a *whole* population, we say the parent operates at population level. Likewise, a parent operates at subpopulation level, individual level, or component level, if the corresponding object is a subpopulation, an individual, or a solution component (or multiple components), respectively.

Operating at component level, a parent optimizer will be used to regulate one or more components of candidate solutions rather than an *entire* solution (individual). Besides, different components of an individual can be evolved by different methods, that is to say, the individual will be evolved by multiple methods simultaneously. In this case, designers may specify some evolution methods for different components or grant them the freedom to opt for their evolution methods according to some indices (e.g., some predefined selection probability). DEPSO-HGH provides an example of the hybrid whose parents operate at component level [see (11)–(13)]. In contrast, each time a parent optimizer works, operating at individual level, it will evolve an entire solution (individual), but usually *not all* solutions in the population. In this case, the individual can *independently* apply the parent optimizer without being required to adopt the same evolution method with any other individuals. The individual may have the freedom to choose its evolution methods according to some selection rules. For example, the individuals in DEPSO-XCPP independently choose their own evolution methods (PSO or DE) according to the success ratio of candidate methods [109].

If a parent optimizer operates at population level, the evolution method once applied will *simultaneously* act on all individuals in the population; howbeit, different methods may work on the whole population alternately at different generations. For example, the parents of DEPSO-ZX operate at population level since its PSO parent evolves the *entire* population and DE follows to further improve the personal best positions of *all* particles [28]. As an *intermediate* case between individual and population levels, subpopulation level indicates a division of the whole population, which is often termed as *multispecies* strategy or *multiswarm* strategy (*aka*, the *niching* technique in GAs research) [136]. Parents that operate at subpopulation level will be applied to different subpopulations. DEPSO-ZZS provides an example of the hybridization at subpopulation level [94]. It is worthwhile to mention that the niching technique is usually applied to *homogeneous* optimizers, such as GAs, giving birth to many niching-based EAs (e.g., the niching GA [136], niching PSO [137], [138], and niching DE [139], [140]). From a broader perspective, hybridization with parents that operate at subpopulation level can be deemed as an *extension* of the niching technique (see also the definition of homogeneity that is given by Talbi [15]). Particularly, as for multiobjective optimization, the parent optimizers can be assigned to different objectives with each parent manipulating its own subpopulation (e.g., DEPSO-GE [90]). The subpopulation level with appropriate division of labor based on specialization of parents is also a nice choice to design a hybrid to solve constrained optimization problems or track multiple optima in multimodal landscapes.

If parents work at the same OL, we term the resultant hybridization as *homogeneous-level* hybridization (HOLH), and the OL of parents is regarded as their HL. In contrast, *heterogeneous-level* hybridization (HELH) means that the OLs of parents are different. For example, regarding the hybridization between a PO and a trajectory search method (TSM), i.e., single-point-based optimizer [17], in which the PO usually operates at *population* level, the TSM will be regarded to be hybridized with the PO at *individual* level if a TSM manipulates only one solution at each generation. As a rule, the HL in HELH between two optimizers is determined by the lower OL of parents. Furthermore, some special HOLH can be termed as *homogeneous-object* hybridization (HOOH), if parents share the *same object* to determine their OLs, which means that they are employed to evolve the same object at their HL. In contrast, *heterogeneous-object* hybridization (HEOH) implies that the objects, which determine the OLs of parents are different even though the objects may be of the same type (e.g., subpopulation type). For example, parents that operate on different subpopulations give birth to HEOH (e.g., DEPSO-ZZS [94]). Besides, undoubtedly we can categorize any HELH as HEOH.

*2) Operating Order:* The OO of parent optimizers refers to the *temporal* assignment of parent optimizers. Parents can work in a *sequential* (*alternate*) or *parallel* order. Talbi terms the sequential OO type as *relay*, meaning that a set of metaheuristics is applied one after another, each using the output of the previous as its input, acting in a *pipeline* fashion [15]. Das categorizes hybrids based on the two OO types as *serial* hybridization and *parallel* hybridization, respectively [144]. Particularly, parents of embedding-based hybrids work together as an *ensemble*, regardless of the OO. Das terms embedding-based hybrids as *implicit* hybridization [144]. For clarity of classification, we term the case of being unable to distinguish the order of parents as "no order".

The OO regarding a hybrid should be canonically analyzed at its HL, which is a crucial rule to identify the OO of parents since different angles of view may lead to different conclusions with respect to OO. For example, the parents in DEPSO-OES1 [103], from a population perspective, hold a parallel OO as they can simultaneously operate on different individuals at each generation; howbeit, they act *alternately* on any single individual from the perspective of their HL (i.e., individual level). To avoid confusion, we will treat the analysis of OO regarding HOOH and HEOH in different ways. For HOOH, parents hold a parallel OO if and only if they can simultaneously operate on the same object that determines their OLs. If the parents evolve the object in a "relay" manner as Talbi claimed [15], we say they hold a sequential OO. For HEOH, parents usually hold a parallel OO as they operate on different objects, facilitating the implementation of parallel OO. However, some special interdependence between parents may force them to carry out a sequential OO. For example, in assistance-based hybrids, the master parent and its assistant *usually* operate on different objects (e.g., in DEPSO-KSSP [117], a population of parameter vectors for DE and a solution population for PSO), and the parents definitely hold a sequential OO since the master relies on the information that is supplied by its assistant to implement its

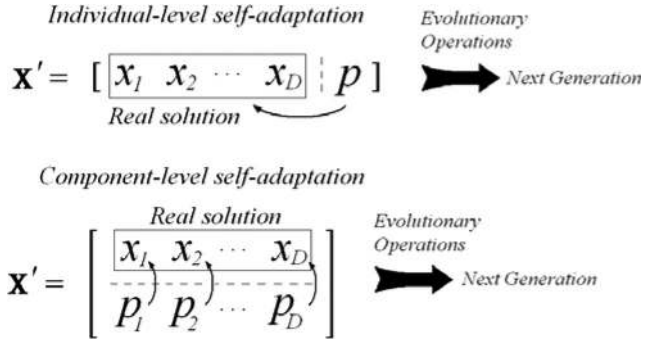



Fig. 2. Illustration of the individual-level self-adaptation and the component-level self-adaptation. The variables $p$ and $p_i$, $i \in \{1, 2, \ldots, D\}$, represent the choice of parent optimizers for the evolution of an individual and a component, respectively. Taking DEPSO for an example, the variables can be set to PSO (0) or DE (1). The extended individual $\mathbf{X}'$ will undergo the evolutionary operations. As opposed to the individual-level self-adaptation, the component-level self-adaptation brings more diversity of evolutionary operations. The quality of the extended individuals is evaluated by the fitness of corresponding real solutions. It is implicitly acknowledged that high-quality solutions come along with appropriate evolution methods.

operations. Another example is the CPSO-$S_K$ that is proposed by Bergh and Engelbrecht [46] in which multiple homogeneous PSOs (parents) operate on different solution components and share them through a context vector for fitness evaluation. The PSO parents will one by one, rather than simultaneously, evolve their respective swarm and update the corresponding solution components of the context vector. The sequential OO, here, is advantageous to restrain different PSOs from the utilization of the same context vector for fitness evaluation and yield a fine-grained search [46].

As for the sequential OO, the alternation of parents, *viz.*, choosing parents to run at different intervals, can be achieved in different ways. Referring to the assortment of alternation, it is worthwhile to mention the classification of adaptive MAs that are proposed by Ong *et al*. [133]. The classification takes into account two criterions to sort different adaptive MAs: *adaptation type* (*static*/*adaptive*/*self-adaptive*) and *adaptation level* (*external*/*local*/*global*). The adaptation type is identified to be *static* if the feedback from the search process regarding the online performance of different memes (i.e., local IMs) is not utilized for the choice of memes. In contrast, both adaptive and self-adaptive types utilize the feedback to guide the choice of memes. However, the self-adaptive type integrates (*encodes*) the choice of memes into the representation of individual solutions. The conceptually extended "individual" involves not only a real solution but also its local IM (meme). Fig. 2 is an illustration of the *individual-level* self-adaptation and the *component-level* self-adaptation. The adaptation level refers to the *extent* of prior knowledge about memes. External-level adaptation usually depends on the experiences of human experts, instead of *online knowledge*, to choose the memes. Local-level and global-level adaptations, which are differentiated by the range of utilized historical knowledge, choose the memes according to their online performance. The aforementioned classification is also applicable to different types of alternation since the alternation of parent optimizers is *de facto* an adaptation behavior. In the following, we provide a subdivision of *static* alternation and *adaptive* alternation to reveal more design issues regarding hybrids based on the sequential OO.

Static alternation can take different forms, such as *deterministic* alternation and *stochastic* alternation. The deterministic alternation means that parents will alternate according to a predefined deterministic order. Therefore, the timetable by which each parent runs and terminates is prepared in advance. For example, in the *two-phase* DEPSO-CNT, PSO runs first, providing DE with the so-called super-fit individual [87]. In DEPSO-ZX, PSO, and DE also alternate *regularly* in a deterministic order [28]. Usually, the deterministic execution order of parents stems from the insight and experience of designers. For example, the idea behind DEPSO-APA is that DE/rand/1/bin is more suitable for early-stage exploration than the basic PSO (Gbest topology), while the latter is more competent for exploitation [88]. Stochastic alternation means that the execution of parents is a *random* event that is dependent on the *selection probability* of different parents, which may be constant or change with evolution stages. The setting of the selection probability that pertains to static alternation is also a pure human factor, which is unrelated to the online performance of parents.

Regarding adaptive alternation, we can adopt various schemes of utilization of feedback information to conduct the selection of parents in the alternation process. The evolution process can switch from one parent to another when it is identified that the in-service parent cannot bring fitness improvement. This scheme seems *conservative* as it does not care about the performance difference of parents. In contrast, a *greedy* scheme only chooses the parent, which performs the best according to search history, resulting in a strong competition among parents. A *moderate* alternative is to adjust the selection probability of parents in light of their relative performance. For example, the PSO and DE in DEPSO-XCPP are adaptively chosen according to their relative success ratio of improvement of the fitness of individuals [109]. To date, how to utilize the feedback from search process to choose competent parents is still an open issue.

For the parallel OO, attention should be paid to the interaction of parents, including the information shared between parents and the occasions on which the information is shared. The parallel order can be implemented with the support of hardware architecture being capable of achieving parallel computing (e.g., single-instruction stream with multiple-data stream and multiple-instruction stream with multiple-data stream). A further discussion on the parallel architecture for the implementation of hybrid metaheuristics can be found in [15].

*3) Type of Information Transfer:* The TIT refers to the interconnection structure of parent optimizers, concerning the information flow between them. By regarding each parent as an "individual (particle)," we can equate the TIT with the population topology of parent optimizers. Fig. 3 provides four typical interconnection structures. Pertaining to the hybridization between *two* parent optimizers, what concerns the classification of the TIT is the direction of information flow, either *unidirectional* (*simplex* TIT) or *bidirectional* (*duplex* TIT). Generally speaking, the duplex TIT leads to a stronger coupling between parents than the simplex TIT. Most existing DEPSOs employ the duplex TIT, manifesting that the *interactive* influence between DE and

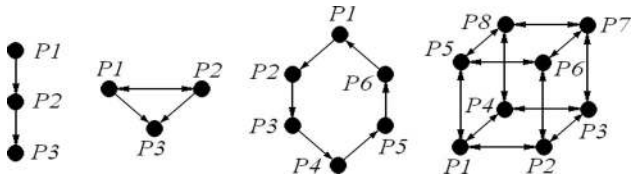



Fig. 3. *Typical interconnection structures*: Arrows represent the direction of the information flows between two parents. A bidirectional arrow indicates that the related parents transfer information in a duplex manner. In the structure that is shown on the far left, Parent P1 works independently since it does not receive any information from the other two parents. The behavior of P2 is affected by P1 as P2 utilizes the information from P1 to guide its search. In the second structure, P1 and P2 interact and jointly guide the search of P3. The third structure looks like the well-known Ring topology in PSO research. Moreover, it defines the information flow between parents, giving birth to a unidirectional flow of the mutual influence of parents. The final cube structure is termed as the island model in niching GA research [15], [136].

PSO has drawn more attention from DEPSO designers. DEPSO-CNT [87], DEPSO-APA [88], and DEPSO-KSSP [117] are the three examples of the simplex TIT. In addition, if the parents in a duplex TIT transfer information of the same type, the TIT will be termed as the *homogeneous* TIT, and *heterogeneous* TIT otherwise.

It should be stressed that the goal of information transfer in hybridization is the information utilization that is linked to interactions between parents. If the transferred information is merely shared but not utilized to affect the search behavior of other parents, the parents will work independently. In this case, the hybridization without interactions cannot bring the expected performance emergence since it can be easily derived that

$$\begin{aligned}\mathrm{Perf}_{AB}(m+n) &= \mathrm{Perf}_A(m) + \mathrm{Perf}_B(n) \\ &\le \max\{\mathrm{Perf}_A(m+n), \mathrm{Perf}_B(m+n)\}\end{aligned}$$

where *AB* is the hybrid of the parents $A$ and $B$, $\mathrm{Perf}_{AB}(m+n)$ is the performance index of *AB* with $m+n$ function evaluations (FEs), and$\mathrm{Perf}_A(m)$ and $\mathrm{Perf}_B(n)$ are the performance indices of $A$ with $m$ FEs and $B$ with $n$ FEs, respectively.

The aforementioned inequality holds for the optimization of any problem, indicating that the hybrid *AB* cannot outperform both parents at the same time. Therefore, the hybridization without interactions is meaningless and the utilization of transferred information is essential. To conclude, *no interaction*, *no hybridization*!

*4) Type of Transferred Information:* As a crucial design element of hybrid optimizers, the TTI is the most complicated factor among the aforementioned hybridization factors. It is not only related to *specific optimizers*, but also to the *type of the problems* to be solved. Common TTIs include *current solutions* (*aka, target vectors* for DE and *current particle positions* for PSO), group of elite solutions, *so-far-best solution* ($\mathbf{gbest}$), *solution components*, and *control parameters*. Regarding PSO, the alternative for the TTI can also be $\mathbf{pbest}$, $\mathbf{nbest}$, or particle *velocities*. For DE, its *base vectors*, *difference vectors*, *mutant vectors,* and *trial vectors* can be the candidate TTIs as well. Besides, the transferred information may be a combination of the TTIs mentioned earlier.

As for multiobjective optimization, the external *archive* that is used to store high-quality (Pareto optimal) nondominated solutions is also a common TTI (e.g., DEPSO-GZQ [89] and DEPSO-GE [90]). For constrained optimization, high-quality feasible and infeasible solutions can be treated as the candidate TTIs. For example, Yang *et al*. proposed a master–slave PSO algorithm to solve constrained optimization problems with two PSOs, a master and a slave, manipulating their own subpopulations and cooperatively updating a feasible leader set (FLS) and an infeasible leader set (ILS) [141]. The FLS and ILS are expected to record elite feasible solutions and high-quality infeasible solutions, respectively. The master PSO and the slave choose $\mathbf{nbest}$ from the FLS and ILS, respectively. Resembling the master–slave PSO, a new DEPSO for constrained optimization can be constructed by enabling DE and PSO to be guided by FLS and ILS, respectively (vice versa).

Some special TTIs can be introduced by specific approaches. For example, in the *coevolutionary* PSO algorithm that is proposed by Krohling and Coelho to solve constraint optimization problems [142], two PSOs operate on real solutions and *Lagrange multipliers*, respectively. The *Lagrange multipliers* are introduced by a *Lagrange* approach, which formulates constrained optimization problems into *min–max* problems. The two PSOs, which are hybridized in a *sequential* OO, share and utilize the real solutions and the *Lagrange* multipliers during fitness evaluations. A similar instance is the hybrid of SA and GA for nonlinear constrained optimization [143].

From the earlier discussion, it can be seen that the TTI is a complicated hybridization factor with plentiful options, making it very hard to enumerate all TTIs. Instead, a coarse-grained classification of the TTI is provided, covering the following TTI types: 1) solutions (e.g., $\mathbf{gbest}$, $\mathbf{pbest}$, $\mathbf{nbest}$, DE's target vectors and trial vectors, the archive of nondominated solutions, and the FLS and ILS mentioned earlier); 2) fitness information; 3) solution components; 4) algorithm-induced in-betweens (e.g., DE's base vectors, mutant vectors, difference vectors, PSO's particle velocities); 5) auxiliaries (e.g., *Lagrange* multipliers); and 6) control parameters.

The difference between solutions and algorithm-induced in-betweens lies in that solutions are expected to have gone through fitness evaluation but algorithm-induced in-betweens are merely intermediate variables that are introduced by specific optimizers and usually taken as ingredients for generation of new solutions.

### *B. Taxonomy*

By the combination of the aforementioned hybridization factors, we build a taxonomy of different hybridization strategies. First of all, we will introduce some concise notations to express the candidate classes with respect to each factor as follows.

1) Parent relationship (PR)
   Collaboration: ⟨C⟩, Embedding: ⟨E⟩, Assistance: ⟨A⟩.
2) Hybridization level (HL)
   Population level: ⟨P⟩, Subpopulation level: ⟨S⟩
   Individual level: ⟨I⟩, Component level: ⟨C⟩.
3) Operating order (OO)
   Sequential order: ⟨S⟩, Parallel order: ⟨P⟩, No order: ⟨N⟩.

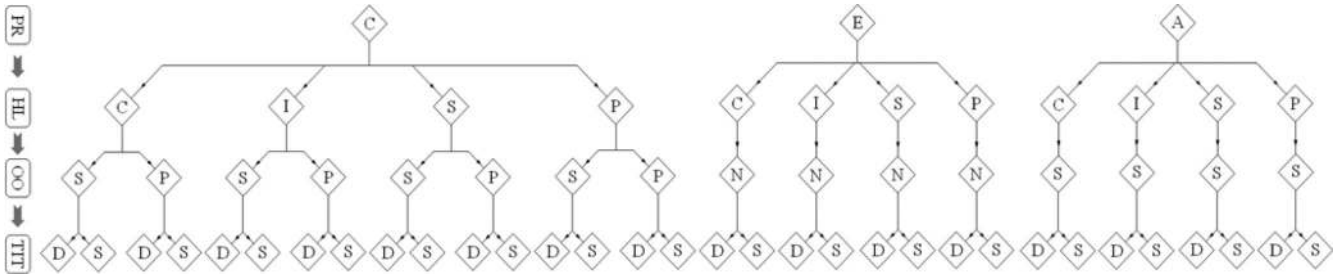



Fig. 4. Taxonomy of hybridization strategies. Each string connecting four hybridization factors corresponds to a class of hybridization strategies.

4) Type of information transfer (TIT)
   Simplex TIT: ⟨S⟩, Duplex TIT: ⟨D⟩.

*Remark 1:* The simple classification of the TIT, here, is only appropriate for hybridization between two parents. For hybridization that involves more than two parents, designers will have more choices with the TIT in respect that the information topology of parents can be flexibly configured in a variety of ways (see Fig. 3). In the generalized case, we can resort to a graph representation of the TIT.

5) Type of transferred information (TTI)
   Solutions: ⟨S⟩, Fitness information: ⟨F⟩, Solution components: ⟨Sc⟩, Auxiliaries: ⟨A⟩, Control parameters: ⟨Cp⟩, Algorithm-induced in-betweens: ⟨Ai⟩.

*Remark 2:* The six TTI types that are mentioned earlier are not *mutually exclusive*, that is to say, a hybridization strategy may involve multiple TTI types simultaneously. For example, solutions ⟨S⟩ are usually transferred along with their fitness information ⟨F⟩. In some special cases, the fitness information may be transferred alone. For brevity, the transferred fitness information will not be labeled in the following TTI classification of different hybrids unless its corresponding solution is not transferred.

Here, we also provide notations for some specific TTIs as follows: current solutions $\langle x \rangle$, group of elite solutions $\langle e \rangle$, $\mathbf{gbest}$ $\langle g \rangle$, $\mathbf{pbest}$ $\langle p \rangle$, $\mathbf{nbest}$ $\langle n \rangle$, base vectors (DE) $\langle b \rangle$, difference vectors (DE) $\langle d \rangle$, mutant vectors $\langle m \rangle$, trial vectors $\langle t \rangle$, particle velocities $\langle v \rangle$, archive of nondominated solutions $\langle a \rangle$, feasible solutions ⟨fs⟩, infeasible solutions ⟨is⟩, Lagrange multipliers ⟨lm⟩.

The five hybridization factors that are mentioned earlier are either fine grained or coarse grained, covering primary elements to be considered in hybridization design and exhibiting a *hierarchical* feature (see Fig. 4). PR is a rough but fundamental classification criterion, grouping hybrids into three general categories—collaboration, embedding, and assistance. The HL that is based on the OLs of parent optimizers helps to differentiate the spatial assignment of parents, while the OO indicates the temporal assignment of parents. Both of them are crucial factors for the design of hybrid optimizers. In contrast, the TIT and TTI are mainly associated with details in hybridization design. With this taxonomy, one can differentiate various hybridization strategies in more details and easily compare different hybridization schemes. Due to the *hierarchical* feature of the taxonomy, one can also take it as a reference of a *step-by-step* procedure to design any hybrid optimization algorithms, by determining in turn PR, HL, OO, TIT, and TTI. The template ⟨PR, HL, OO, TIT⟩ that incorporates the first four factors will be utilized to represent various sorts of hybridization strategies. The reason that the TTI is not included in this template is that different TTI types in the classification of the TTI are not mutually exclusive for the construction of a hybridization strategy. In the proposed taxonomy, the TTI will be treated as a *complementary* criterion for the detailed analysis of or to design a hybridization strategy.

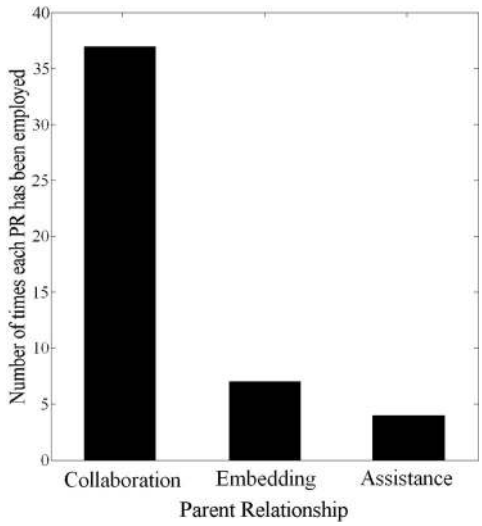



Fig. 5. Number of times each PR has been employed among all DEPSOs that are surveyed.

According to the classification of each factor in the template, there seem to be total $3 \times 4 \times 3 \times 2 = 72$ classes of hybridization strategies. However, there exist some special connections (constraints) between the hybridization factors, which make it impossible to arbitrarily combine these factors. For example, parents in embedding- or assistance-based hybridization cannot hold a parallel OO (see Section IV-A).

According to the taxonomy, existing DEPSOs are sorted in Table I with their parents, applications, and inventors' ideas provided. The frequencies of three PRs being employed among all DEPSOs that are surveyed are compared in Fig. 5, from which it can be seen that the collaboration PR was widely employed by DEPSO designers. In contrast, fewer DEPSOs hold the embedding PR or assistance PR. Furthermore, the distribution of existing collaboration-based DEPSOs at four kinds of HLs is shown in Fig. 6. According to the statistical results, the population level has received more attention, while the other three HLs

TABLE I
CLASSIFICATION OF EXISTING DEPSOS

| DEPSO | Parents | Hybridization strategy | Inventors' ideas * | Applications |
|---|---|---|---|---|
| DEPSO-H [72] | DE/rand/1/bin, PSO (Gbest) | <C,P,S,D>, HOLH, HOOH, deterministic static alternation, homogeneous TIT, TTI: <S>, $PSO \underset{\langle x\rangle}{\overset{\langle x\rangle}{\rightleftharpoons}} DE$ | DE is run to move particles from a poorer area to a better one | Only through benchmark test |
| DEPSO-RJAMB [73] | A hybrid (memetic) DE, Classic PSO (Gbest) | <C,P,S,D>, HOLH, HOOH, deterministic static alternation, heterogeneous TIT, TTI: <S>, $PSO \underset{\langle p\rangle,\langle g\rangle}{\overset{\langle x\rangle}{\rightleftharpoons}} DE$ | Combine the vibrancy and explorative nature of PSO with the superior exploitative nature of a hybrid DE | Economic dispatch [73] |
| DEPSO-ZX [28] | DE/best/2/bin, Classic PSO (Gbest) | <C,P,S,D>, HOLH, HOOH, deterministic static alternation, homogeneous TIT, TTI: <S>, $PSO \underset{\langle p\rangle,\langle g\rangle}{\overset{\langle p\rangle,\langle g\rangle}{\rightleftharpoons}} DE$ | Use DE to perturb the personal best positions of particles | Multimodal image registration [74], Modeling of gene regulatory networks [75], Structure optimization of a high-temperature super conducting cable [76], Design of FIR filters [77] |
| DEPSO-MV [78] | DE/rand/2/bin, A discrete PSO (Ring topology) | <C,P,S,D>, HOLH, HOOH, deterministic static alternation, homogeneous TIT, TTI: <S>, $PSO \underset{\langle p\rangle,\langle g\rangle}{\overset{\langle p\rangle,\langle g\rangle}{\rightleftharpoons}} DE$ | Use DE to perturb the personal best positions of particles | Design of combinatorial logic circuit (multi-objective optimization) [78] |
| DEPSO-LWJH [79] | DE/current-to-best/1/bin, Classic PSO (Gbest) | <C,P,S,D>, HOLH, HOOH, deterministic static alternation, homogeneous TIT, TTI: <S>, $PSO \underset{\langle p\rangle,\langle g\rangle}{\overset{\langle p\rangle,\langle g\rangle}{\rightleftharpoons}} DE$ | Balance the exploration and exploitation abilities by combining DE and PSO. The inventors also employ chaotic local search to improve the local exploitation of the DEPSO | Training of artificial neural networks [79] |
| DEPSO-HWLR [80] | DE/best/1/bin, Classic PSO (Gbest) | <C,P,S,D>, HOLH, HOOH, deterministic static alternation, homogeneous TIT, TTI: <S>, $PSO \underset{\langle p\rangle,\langle g\rangle}{\overset{\langle p\rangle,\langle g\rangle}{\rightleftharpoons}} DE$ | Enhance particle's diversity through the combination of PSO and DE operators | Back analysis of mechanics parameters in engineering projects [80] |
| DEPSO-LCW [81] | DE/rand/1 <DE1>, DE/rand/2 <DE2>, DE/current-to-best/1 <DE3>, Classic PSO (Gbest) | <C,P,S,D>, HOLH, HOOH, deterministic static alternation, heterogeneous TIT, TTI: <S>, $PSO \underset{\langle p\rangle}{\overset{\langle x\rangle}{\rightleftharpoons}} DE1$, $PSO \underset{\langle p\rangle}{\overset{\langle x\rangle}{\rightleftharpoons}} DE2$, $PSO \underset{\langle p\rangle}{\overset{\langle x\rangle,\langle g\rangle}{\rightleftharpoons}} DE3$ | DE is expected to force PSO jump out of stagnation. The application of multiple differential mutation strategies contributes to the diversity of evolution operations. | Constrained numerical and engineering optimization [81] |
| DEPSO-XXW [82] | DE/current/2/bin, Classic PSO (Gbest) | <C,P,S,D>, HOLH, HOOH, deterministic static alternation, homogeneous TIT, TTI: <S>, $PSO \underset{\langle p\rangle}{\overset{\langle p\rangle}{\rightleftharpoons}} DE$ | DE follows PSO in each generation to further improve the personal best positions of particles. | Partitional analysis [82] |
| DEPSO-NZL [86] | DE/best/2/bin, classic PSO (Gbest) | <C,P,S,D>, HOLH, HOOH, deterministic static alternation, heterogeneous TIT, TTI: <S>, $PSO \underset{\langle x\rangle}{\overset{\langle g\rangle,\langle x\rangle}{\rightleftharpoons}} DE$ | Introduce differential mutation into the basic PSO to alleviate the premature convergence problem | Training of feed-forward neural network [86] |
| DEPSO-GZQ [89] | DE/best/2/bin, PSO (Gbest) (time-variant acceleration coefficients) (velocity removed) | <C,P,S,D>, HOLH, HOOH, deterministic static alternation, homogeneous TIT, TTI: <S>, $PSO \underset{\langle a\rangle}{\overset{\langle a\rangle}{\rightleftharpoons}} DE$ | Synergy of PSO's exploration ability and the exploitation ability of DE/best/2/bin. DE is employed to exploit the subspace with sparse solutions | Multi-objective environment/economic power dispatch [89] |
| DEPSO-JC [91] | DE/rand/1/bin, Classic PSO (Gbest) | <C,P,S,D>, HOLH, HOOH, deterministic static alternation, homogeneous TIT, TTI: <S>, $PSO \underset{\langle a\rangle}{\overset{\langle a\rangle}{\rightleftharpoons}} DE$ | DE has powerful capacity to scatter the feasible space, enhance the diversity of archive and provide good guide information to particles. | Only through benchmark test (multi-objective optimization) |

* The inventors' ideas are principally extracted from related literature.

relatively lack the concern of researchers. The distribution of existing DEPSOs on nine hybridization-strategy classes involved is shown in Fig. 7. Among all classes of hybridization strategies that are involved in the proposed taxonomy, $\langle$C,P,S,D$\rangle$ is the most favored. Besides, $\langle$C,S,P,D$\rangle$ and $\langle$C,I,S,D$\rangle$ also have gained more concern from researchers. On one hand, this can reflect the popularity and potential advantages of these hybridization strategies. On the other hand, however, the unbalanced situation may conceal the potential superiority of some other promising strategies. On the whole, there are more hybridization-strategy classes that are involved in the proposed taxonomy but not reported in the DEPSO literature, such as $\langle$C,C,P,D$\rangle$, $\langle$C,I,P,D$\rangle$, $\langle$C,P,P,D$\rangle$, $\langle$E,I,N,D$\rangle$, and $\langle$A,I,S,D$\rangle$.

The proposed taxonomy can be easily applied to the analysis of hybridization strategies regarding various hybrid optimizers. In the following, we utilize the proposed taxonomy and related notations to classify several representative hybrids that are reported in frequently cited works. In addition to the classification, the main ideas in each hybrid are also summarized as a reference for interested researchers.

*1) HGAPSO (hybrid of genetic algorithm and particle swarm optimization) Proposed by Juang [20]:* GA$\oplus$PSO$\langle$C,P,S,D$\rangle$: [HOLH], [HOOH], [deterministic static alternation], [heterogeneous TIT], [TTI: $\langle$S$\rangle$, $\mathrm{PSO} \underset{\langle e\rangle}{\overset{\langle p\rangle}{\rightleftharpoons}} \mathrm{GA}$].

*Inventor's ideas:* Elite individuals in a generalized population, which occupy half of the population, will be evolved by PSO. The GA operates on the evolved elites to generate some offspring. The enhanced individuals that are generated by PSO and the GA offspring will compete through a ranking selection, and only the better half of the generalized population (PSO-enhanced elites plus GA-generated offspring) will enter the next generation. Note that the HL of this hybrid is regarded as population level since the worse half of the generalized population will be always eliminated and the better half can be viewed as a whole population for both GA and PSO.

*2) GQL (hybrid of Genetic Algorithm and Quasi-Newton Inspired by Lamarckian Evolution) and GQD (hybrid of Genetic Algorithm and Quasi-Newton Inspired by Darwinian Evolution) Proposed by Renders and Flasse [21]:* GA$\oplus$QN(quasi-Newton method).

TABLE I
(*Continued.*)

| DEPSO | Parents | Hybridization strategy | Inventors' ideas | Applications |
|---|---|---|---|---|
| DEPSO-LHTC [84] | DE/rand/1/bin, PSO with a linearly decreasing inertia weight (Gbest) | <C,P,S,D>, HOLH, HOOH, deterministic static alternation, homogeneous TIT, TTI: <S>, $PSO \underset{\langle x\rangle}{\overset{\langle x\rangle}{\rightleftharpoons}} DE$ | DE changes particles' positions with the goal of keeping particles' diversity. | Multistage inventory control in supply chain management [84], Supply chain production planning [125] |
| DEPSO-XG [85] | DE/current/1/bin (plus a prior crossover to diversify the population), a modified PSO (Gbest) | <C,P,S,D>, HOLH, HOOH, deterministic static alternation, homogeneous TIT, TTI: <S>, $PSO \underset{\langle x\rangle}{\overset{\langle x\rangle}{\rightleftharpoons}} DE$ | DE is expected to enhance the exploitation in the neighborhood of current particle positions. *Average position and velocity of particles incorporated into basic PSO to concern the evolution effect of the whole swarm. | Reactive power optimization [126] |
| DEPSO-DAK [105] | DE/rand/1/bin, Classic PSO (Gbest) | <C,P,S,D>, HOLH, HOOH, stochastic static alternation, homogeneous TIT, TTI: <S>, $PSO \underset{\langle x\rangle}{\overset{\langle x\rangle}{\rightleftharpoons}} DE$ | DE is employed as operators of PSO to improve the search speed and exploitation. | Design of beta basis function neural networks [105] |
| DEPSO-CNT [87] | DE/rand/1/bin, Basic PSO without inertia weight (Gbest) | <C,P,S,S>, HOLH, HOOH, deterministic static alternation, TTI: <S>, $PSO \xrightarrow{\langle x\rangle} DE$ | PSO is expected to provide DE with high-quality initial solutions. Two local searchers are used to assist DE in a meta-Lamarckian logic. | optimal drive design for a direct current motor, the design of a digital filter [87] |
| DEPSO-APA [88] | DE/rand/1/bin, PSO (Gbest) with linearly decreasing inertia weight | <C,P,S,S>, HOLH, HOOH, deterministic static alternation, TTI: <S>, $DE \xrightarrow{\langle e\rangle} PSO$ | The initial search space is contracted by DE and then PSO is applied to further refine the search space. | Only through benchmark test |
| DEPSO-NL [92] | DE/rand/1/bin, Classic PSO (Gbest) | <C,S,P,D>, HOLH, HEOH, homogeneous TIT, TTI: <S>, $PSO \underset{\langle g\rangle}{\overset{\langle g\rangle}{\rightleftharpoons}} DE$ | The information transferred among individuals of different population will help the individuals to avoid misjudging information and becoming trapped by poor local minima. | automatic design of fuzzy identifier for a nonlinear dynamic system [93] |
| DEPSO-ZZS [94] | DE/current-to-best/1/bin, Classic PSO (Gbest) | <C,S,P,D>, HOLH, HEOH, homogeneous TIT, TTI: <S>, $PSO \underset{\langle g\rangle}{\overset{\langle g\rangle}{\rightleftharpoons}} DE$ | Co-evolution of DE and PSO swarms sharing global best information. Chaotic perturbation employed to improve search speed | Only through benchmark test |
| DEPSO-WYZ [95] | DE1,DE2: two of three candidate DE variants Classic PSO (Gbest) | <C,S,P,D>, HOLH, HEOH, homogeneous TIT, TTI: <S>, $PSO \underset{\langle g\rangle}{\overset{\langle g\rangle}{\rightleftharpoons}} DE1$, $PSO \underset{\langle g\rangle}{\overset{\langle g\rangle}{\rightleftharpoons}} DE2$, $DE1 \underset{\langle g\rangle}{\overset{\langle g\rangle}{\rightleftharpoons}} DE2$ | Combine the local optimization ability of PSO and the global optimization ability of DE to gain more optimization precision and search efficiency. | Only through benchmark test |
| DEPSO-SC [96] | DE/rand/1/bin, basic PSO, GA | <C,S,P,D>, HOLH, HEOH, homogeneous TIT, TTI: <S>, $PSO \underset{\langle g\rangle}{\overset{\langle g\rangle}{\rightleftharpoons}} DE$, $PSO \underset{\langle g\rangle}{\overset{\langle g\rangle}{\rightleftharpoons}} GA$, $GA \underset{\langle g\rangle}{\overset{\langle g\rangle}{\rightleftharpoons}} DE$ | The cooperative procedure runs the three basic algorithms in parallel, while letting the best individuals migrate to the other populations at prescribed intervals. | Spacecraft trajectory optimization [96],[97] |
| DEPSO-HW [100] | DE/rand/1/bin, Classic PSO (Gbest) | <C,S,P,D>, HOLH, HEOH, homogeneous TIT, TTI: <S>, $PSO \underset{\langle x\rangle}{\overset{\langle x\rangle}{\rightleftharpoons}} DE$ | The hybrid stresses the balance between PSO's quick convergence and DE's preserving diversity. PSO and DE perform exploration and several local searches perform exploitation | Open vehicle routing problem solving [100] |
| DEPSO-WGHZ [101] | DE: one of four cultured DE variants, PSO with a linearly decreasing inertia weight (Gbest) | <C,S,P,D>, HOLH, HEOH, homogeneous TIT, TTI: <S>, $PSO \underset{\langle x\rangle}{\overset{\langle x\rangle}{\rightleftharpoons}} DE$ | The mutation operator in DE within the culture algorithm framework is introduced to increase the diversity of PSO swarm. | Only through benchmark test |

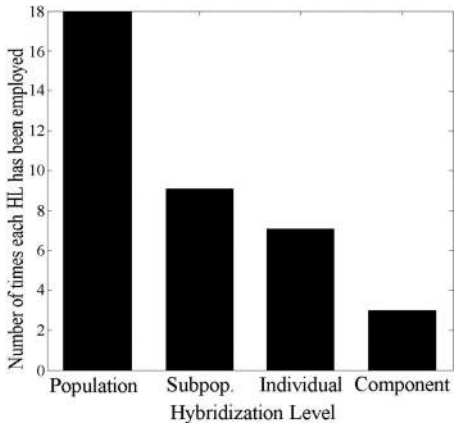



Fig. 6. Number of times each HL has been employed among all collaboration-based DEPSOs that are surveyed.

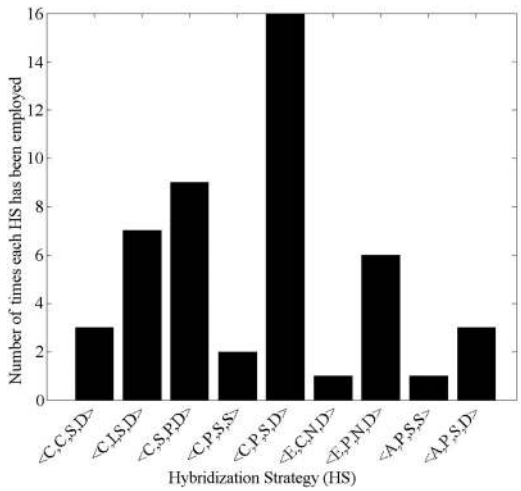



Fig. 7. Number of times each involved hybridization strategy has been employed among all DEPSOs that are surveyed.

*a) GQD (Darwin-inspired hybrid):* ⟨C,P,S,D⟩: [HOLH], [HOOH], [deterministic static alternation], [heterogeneous TIT], [TTI: ⟨S⟩&⟨F⟩, GA $\underset{\langle F\rangle}{\overset{\langle x\rangle}{\rightleftharpoons}}$ QN].

*Inventors' ideas:* The GA provides QN with starting points, and the fitness of any GA individual will be evaluated by that of its corresponding enhanced individual that is generated by QN. However, the GA will operate on the original individuals rather than the improved ones. The hybridization strategy is termed as *Baldwinian* learning in the research of MAs. Although QN is, in nature, a TSM, we identify the OL of GQD to be the population level (⟨P⟩) since all individuals in each generation will go through the improvement of QN.

*b) GQL (Lamark-inspired hybrid):* ⟨C,P,S,D⟩: [HOLH], [HOOH], [deterministic static alternation], [homogeneous TIT], [TTI: ⟨S⟩, GA $\underset{\langle x\rangle}{\overset{\langle x\rangle}{\rightleftharpoons}}$ QN].

TABLE I
(*Continued.*)

| DEPSO | Parents | Hybridization strategy | Inventors' ideas | Applications |
|---|---|---|---|---|
| DEPSO-GE [90] | DE/rand/1/bin, Classic PSO (Gbest) | <C,S,P,D>, HOLH, HEOH, homogeneous TIT, TTI: <S>, $PSO \underset{<a>}{\overset{<a>}{\rightleftharpoons}} DE$ | Co-evolution of DE and PSO swarms: each swarm is assigned one objective to optimize | Only through benchmark test (multi-objective optimization) |
| DEPSO-YMN [98] | DE/rand/2/bin, Quantum-behaved PSO (Gbest) | <C,S,P,D>, HOLH, HEOH, heterogeneous TIT, TTI: <S>, $PSO \underset{<g>,<p>}{\overset{<x>}{\rightleftharpoons}} DE$ | In order to keep population diversity, the mechanism of cultural algorithm is used to hybridize DE with PSO. DE evolves elite solutions and PSO evolves the rest. | Only through benchmark test |
| DEPSO-PTCY [99] | SaNSDE, Classic PSO (Gbest) | <C,S,P,D>, HOLH, HEOH, homogeneous TIT, TTI: <S>, $PSO \underset{<e>}{\overset{<e>}{\rightleftharpoons}} DE$ | Instead of betting the entire time budget on a single algorithm, such a risk can be reduced by distributing the time budget to multiple algorithms. | Only through benchmark test |
| DEPSO-HGH [39] | DE/mid-to-better/1/bin, Classic PSO (Gbest) | <C,C,S,D>, HOLH, HOOH, stochastic static alternation, homogeneous TIT, TTI: <Sc>, $PSO \underset{<Sc>}{\overset{<Sc>}{\rightleftharpoons}} DE$ | PSO is incorporated into DE in order to maintain diversity and explore the search space more efficiently. | Only through benchmark test |
| DEPSO-DRKC [102] | DE/rand/1/arith, Basic PSO with random inertia weight (Gbest) | <C,C,S,D>, HELH (DE's OL: <C>, PSO's OL: <P>), HEOH, deterministic static alternation, homogeneous TIT, TTI: <Sc,Ai>, $PSO \underset{<Sc>,<i>}{\overset{<Sc>,<i>}{\rightleftharpoons}} DE$ | DE is employed to refine current particle positions by tuning the selected dimensions of the positions. | Maximization of camera coverage area in the coordination of robot ants [102] |
| DEPSO-EPV [83] | DE: one of six candidate variants, PSO: Lbest(Ring) or Gbest | <C,I,S,D>, HELH (DE's OL: <I>, PSO's OL:<P>), HEOH, adaptive alternation, homogeneous TIT, TTI: <S>, $PSO \underset{<p>}{\overset{<p>}{\rightleftharpoons}} DE$ | The selective evolution of the particle personal experience by DE will benefit a good balance between exploration and exploitation | Only through benchmark test |
| DEPSO-OES1 [103] | DE/rand/1/bin, Classic PSO (Gbest) | <C,I,S,D>, HOLH, HOOH, stochastic static alternation, homogeneous TIT, TTI: <S>, $PSO \underset{<g>}{\overset{<g>}{\rightleftharpoons}} DE$ | Enhance the diversity of evolution methods for each individual by employing both DE and PSO. | Only through benchmark test |
| DEPSO-KL [104] | DE/current/1/bin, Classic PSO (Gbest) | <C,I,S,D>, HOLH, HOOH, stochastic static alternation, heterogeneous TIT, TTI: <S>, $PSO \underset{<x>}{\overset{<x>,<p>}{\rightleftharpoons}} DE$ | Enhance the diversity of evolution methods for each individual by employing both DE and PSO. Q-learning is employed to regulate control parameters.. | Structure optimization of 3-bar planar truss, pistol oil problem [104] |
| DEPSO-PTGA [106] | DE/rand/1/bin, Classic PSO (Gbest) | <C,I,S,D>, HELH (DE's OL: <P>, PSO's OL: <I>), HEOH, conservative adaptive alternation, homogeneous TIT, TTI: <S>, $PSO \underset{<x>}{\overset{<x>}{\rightleftharpoons}} DE$ | The inclusion of PSO creates a perturbation and helps in maintaining population diversity and producing high-quality solutions. | Only through benchmark test |
| DEPSO-KWJ [107] | One of the four DE versions presented in [107], Classic PSO (Gbest) | <C,I,S,D>, HELH (DE's OL:<I>, PSO's OL:<P>), HEOH, conservative adaptive alternation, homogeneous TIT, TTI: <S>, $PSO \underset{<x>}{\overset{<x>}{\rightleftharpoons}} DE$ | The mutation operators of DE are used for improving the diversity and exploration of PSO. | Economic dispatch [107] |
| DEPSO-TH [108] | Basic DE\PSO\GA | <C,I,S,D>, HOLH, HOOH, adaptive alternation, homogeneous TIT, TTI: <S>, $PSO \underset{<x>}{\overset{<x>}{\rightleftharpoons}} DE$ , $PSO \underset{<x>}{\overset{<x>}{\rightleftharpoons}} GA$ , $GA \underset{<x>}{\overset{<x>}{\rightleftharpoons}} DE$ | Each individual chooses its evolution method according to the evaluated value of each candidate method. | Only through benchmark test |
| DEPSO-XCPP [109] | DE/rand/1/bin, Classic PSO (von Neumann topology) | <C,I,S,D>, HOLH, HOOH, moderate adaptive adaptation, homogeneous TIT, TTI: <S>, $PSO \underset{<p>}{\overset{<p>}{\rightleftharpoons}} DE$ | On-the-fly adaptation of evolution methods for individuals in a statistical learning way. | Only through benchmark test |
| DEPSO-DKC [110] | DE/none/1/none, PSO (Gbest) without the individual cognition term | <E,P,N,D>, HOLH, HOOH, heterogeneous TIT, TTI: <S,Ai>, $PSO \underset{<d>*}{\overset{<x>}{\rightleftharpoons}} DE$ * DE's mutation is abandoned. The primary feature of DE in this hybrid is the difference of randomly selected individuals used as a perturbation of particle velocities. | The difference of randomly selected individuals is used to perturb particle velocities | spatial clustering [111], fault diagnosis of gear-box [112] |

*Inventors' ideas:* The GA operates on the enhanced individuals that are improved by QN and provides QN with initial points. As opposed to GQD, the improvement of individuals in GQL is inherited into the chromosomes in the GA. The hybridization strategy is termed as *Lamarckian* learning in the research of MAs. Both GQD and GQL are expected to combine the advantage of the GA in exploration with that of QN in exploitation.

*3) MOGLS (Multi-Objective Genetic Local Search) Proposed by Ishibuchi and Murata [22]:* GA$\oplus$LS$\langle$C,P,S,D$\rangle$: [HOLH], [HOOH], [deterministic static alternation], [homogeneous TIT], [TTI:$\langle$S$\rangle$, $\mathrm{GA} \underset{\langle x \rangle}{\overset{\langle x \rangle}{\rightleftharpoons}} \mathrm{LS}$].

*Inventors' ideas:* A truncated LS is employed with the goal to alleviate the great burden of computation-intensive local exploitation and to achieve a better tradeoff between exploration and exploitation (Tr:Er&Ei). The LS is implemented by a shift mutation to solve the multiobjective flowshop scheduling problem. Except for some special techniques to handle multiple objectives (e.g., a random weighting technique) and the truncated LS, the MOGLS shares the same idea with the GQL that is proposed by Renders and Flasse [21]. In fact, both of them belong to the family of MAs.

*4) CPSO (Chaotic Particle Swarm Optimization) Proposed by Liu et al. [23]:* PSO$\oplus$CS (chaotic search) $\langle$C,I,S,D$\rangle$: [HELH (PSO's OL: $\langle$P$\rangle$, CS's OL: $\langle$I$\rangle$)], [HEOH], [deterministic static alternation], [homogeneous TIT], [TTI: $\langle$S$\rangle$, $\mathrm{PSO} \underset{\langle g \rangle}{\overset{\langle g \rangle}{\rightleftharpoons}} \mathrm{CS}$].

*Inventors' ideas:* Chaotic dynamics is incorporated into a PSO with adaptive inertia weight to enrich the searching behavior and avoid being trapped into local optimum. The global best particle in PSO is further improved and updated by CS. Note that the chaotic search is *de facto* an explorative TSM as it does not exploit local information. Liu *et al*. also employed a **gbest**-centered space contraction strategy to accelerate the exploration. Besides, in each generation, only a group of elite solutions will enter the new generation with the other solutions

TABLE I
(*Continued.*)

| DEPSO | Parents | Hybridization strategy | Inventors' ideas | Applications |
|---|---|---|---|---|
| DEPSO-L [113] | DE/nbest/1'/none [1': the difference vector is replaced by individual cognition, see (6)] PSO: Ring topology, velocity | <E,P,N,D>, HOLH, HOOH, homogeneous TIT, TTI: <S,Ai>, $PSO \underset{\langle n\rangle,\langle p\rangle,\langle x\rangle,\langle v\rangle}{\overset{\langle n\rangle,\langle p\rangle,\langle x\rangle,\langle v\rangle}{\rightleftharpoons}} DE$ * PSO's original operations are abandoned. The features of PSO in this hybrid include PSO's population topology and velocity in the form of a derivative of position. | Incorporate the mechanisms of PSO, including velocity and population topology, into DE | Tension/compression string design, economic dispatch [113] |
| DEPSO-SL [114] | DE/**nbest**/1/bin, PSO: Ring topology, velocity | <E,P,N,D>, HOLH, HOOH, homogeneous TIT, TTI: <S,Ai>, $PSO \underset{\langle n\rangle,\langle x\rangle,\langle v\rangle}{\overset{\langle n\rangle,\langle x\rangle,\langle v\rangle}{\rightleftharpoons}} DE$ * PSO's original operations are abandoned. The features of PSO in this hybrid include PSO's population topology and velocity in the form of a derivative of position. | Individuals are created, not only by crossover and mutation operation as in DE, but also by PSO (velocity mechanism and population topology). | Only through benchmark test |
| DEPSO-DACK [115] | A combination of DE/current-to-**nbest**/1/bin and DE/current-to-best/1/bin, PSO: Ring topology | <E,P,N,D>, HOLH, HOOH, heterogeneous TIT, TTI: <S>, $PSO \underset{\langle x\rangle}{\overset{\langle n\rangle}{\rightleftharpoons}} DE$. * The unique feature of PSO retained in this hybrid is the population topology. The update of **nbest** based on solutions generated by DE is regarded as a PSO behavior. | Introduce the population topology of PSO into DE/current-to-best/1/bin to favor a better tradeoff between exploration and exploitation | Spread spectrum radar poly-phase code design, parameter estimation for frequency-modulated sound waves [115], Kernel-induced fuzzy clustering of image pixels [124] |
| DEPSO-XLFWW [116] | DE/current/1'/bin (1': the difference vector is replaced by social learning, see (6)), PSO: Gbest topology, social learning | <E,P,N,D>, HOLH, HOOH, heterogeneous TIT, TTI: <S>, $PSO \underset{\langle x\rangle}{\overset{\langle g\rangle}{\rightleftharpoons}} DE$. * The PSO features retained in this hybrid include the population topology and social learning in PSO. The update of **gbest** based on solutions generated by DE is regarded as a PSO behavior. | Generate the offspring according to the solution's own experience and the experience of neighbors. | Only through benchmark test |
| DEPSO-KSSP [117] | DE/rand/1/bin, Classical PSO (Gbest) | <A,P,S,S>, HOLH, HEOH, deterministic static alternation, TTI: <Cp>, $PSO \xrightarrow{\langle Cp\rangle} DE$ | DE is employed for the parameter selection of PSO | Generation expansion planning [117] |
| DEPSO-OES2 [103] | DE/bbpso/1/bin, BBPSO (Gbest) | <A,P,S,D>, HOLH, HOOH, deterministic static alternation, heterogeneous TIT, TTI: <S,Ai>, $PSO \underset{\langle x\rangle}{\overset{\langle b\rangle}{\rightleftharpoons}} DE$ | Use the point generated by BBPSO as the base vector for DE's mutation | unsupervised image classification [119] |
| DEPSO-JWJ [118] | DE/current/2/bin, Claasical PSO (Gbest) <PSO1>, BBPSO (Gbest) <PSO2> | <A,P,S,D>, HOLH, HEOH, conservative adaptive alternation, homogeneous TIT, TTI: <Ai>, $PSO \underset{\langle v\rangle}{\overset{\langle v\rangle}{\rightleftharpoons}} DE$, $PSO = PSO1 \oplus PSO2$ ($\oplus$ : hybridized with), $PSO1 \underset{\langle x\rangle}{\overset{\langle p\rangle,\langle g\rangle}{\rightleftharpoons}} PSO2$ | DE is employed to regulate particle velocities when PSO is trapped into stagnation | Only through benchmark test |
| DEPSO-WL [120] | DE/rand/1/bin, Classical PSO (Gbest) | <A,P,S,D>, HOLH, HOOH, deterministic static alternation, heterogeneous TIT, TTI: <S>, $PSO \underset{\langle g\rangle}{\overset{\langle x\rangle}{\rightleftharpoons}} DE$ | DE chooses the leaders (**gbest**) for PSO in multi-objective optimization | Only through benchmark test |
| DEPSO-ZNLOD [121] | DE/nbest/1/bin, Classical PSO (Neighborhood is determined by a predefined ball region) | <E,P,N,D> (HOOH), <C,P,S,D> (HOOH: DEPSO $\oplus$ PSO, deterministic static alternation), HOLH, heterogeneous TIT, TTI: <S,Ai>, $PSO \underset{\langle x\rangle,\langle v\rangle,\langle p\rangle}{\overset{\langle n\rangle,\langle p\rangle,\langle x\rangle}{\rightleftharpoons}} DE$ | Modified DE operators are used to enhance each particle's local tuning ability. Besides, a random moving strategy is proposed to enhance the algorithm's exploration. | Geometric constraint solving [123] |
| DEPSO-LWY [122] | DE/rand/1/bin (perturbed by velocity), PSO(Gbest):velocities and positions are updated according to the knowledge in belief space. | <E,C,N,D> (HEOH), <C,C,S,D> (HEOH, stochastic static alternation), <C,P,S,D> (HOOH: DEPSO $\oplus$ PSO, deterministic static alternation), HELH (DE's OL: <C>, PSO's OL: <P>), heterogeneous TIT, TTI: <S,Ai>, $PSO \underset{\langle x\rangle}{\overset{\langle v\rangle,\langle x\rangle}{\rightleftharpoons}} DE$ | Introduce the differential variation mechanism into PSO to maintain swarm diversity. The operations of PSO are redesigned in the framework of cultural algorithm (knowledge utilization in belief space). | Only through benchmark test |

being replaced by new solutions that are randomly generated within contracted space.

*5) MA-S1 and MA-S2 Proposed by Ong and Keane [24]:* GA$\oplus$multiple LSs $\langle$C,I,S,D$\rangle$: [HELH (GA's OL: $\langle$P$\rangle$, each LS's OL: $\langle$I$\rangle$)], [HEOH], [deterministic static alternation], [homogeneous TIT], [TTI: $\langle$S$\rangle$, $\mathrm{GA} \underset{\langle x\rangle}{\overset{\langle x\rangle}{\rightleftharpoons}} \{\mathrm{LS}_1, \mathrm{LS}_2, \ldots, \mathrm{LS}_m\}$].

*Inventors' ideas:* The two MAs are *de facto* two frameworks to hybridize multiple LS methods (memes) with the GA in a meta-Lamarckian manner, which is primarily related to the topic of how to choose LSs according to their online performance. The core of MA-S1 is a subproblem decomposition strategy, which uses a distance metric to define the neighborhood of any individual. The LS for a given individual will be chosen from those employed by its neighbors according to their performance. In contrast, MA-S2 employs a biased roulette wheel scheme to bias the choice of LSs by accumulated knowledge on the performance of each LS candidate.

*6) GASA Proposed by Wang and Zheng [25]:* GA$\oplus$SA (simulated annealing) $\langle$C,P,S,D$\rangle$: [HOLH], [HOOH], [deterministic static alternation], [homogeneous TIT], [TTI:$\langle$S$\rangle$, $\mathrm{GA} \underset{\langle x\rangle}{\overset{\langle x\rangle}{\rightleftharpoons}} \mathrm{SA}$].

*Inventors' ideas:* The GA provides SA with a set of initial solutions at each scheduled temperature. SA performs a Metropolis sampling for each solution until a predefined equilibrium condition is reached. The GA uses the solutions that are found by the SA to continue its parallel evolution. In fact, the SA here plays the role of an IM, carrying out a neighborhood search, though it has excellent exploration capability.

Another well-known hybrid of the GA and SA was proposed by Mahfoud and Goldberg [145]. In this GASA, a temperature-regulated selection that originates from SA operates on parent and offspring solutions in the GA to choose the solutions for the next generation. Therefore, the SA in essence is embedded as a selection mechanism into the GA. Accordingly, this GASA can be categorized into ⟨E,P,N,D⟩.

*7) Simplex-GA Proposed by Yen et al. [26]:* GA⊕CPS (concurrent probabilistic simplex), ⟨C,S,P,D⟩: [HELH (GA's OL:⟨P⟩, CPS's OL:⟨S⟩)], [HEOH], [heterogeneous TIT], [TTI:⟨S⟩, $\mathrm{GA} \underset{\langle x \rangle}{\overset{\langle e \rangle}{\rightleftharpoons}} \mathrm{CPS}$].

*Inventors' ideas:* The hybrid implies a hierarchical fitness-ranking-based division of the whole population. The GA evolves the whole population, and meanwhile, CPS operates on a group of top-ranking solutions. Besides, a part of the best solutions in the CPS subpopulation directly enter the next generation. The CPS is an exploitative optimizer that is expected to search the local space around some high-quality solutions.

*8) DE/EDA Proposed by Sun et al. [27]:* EDA⊕DE (DE/mid-to-better/1/bin), ⟨C,C,S,D⟩: [HOLH], [HOOH], [stochastic static alternation], [heterogeneous TIT], [TTI:⟨S⟩, ⟨Sc⟩, $\mathrm{EDA} \underset{\langle e \rangle, \langle Sc \rangle}{\overset{\langle Sc \rangle}{\rightleftharpoons}} \mathrm{DE}$].

*Inventors' ideas:* DE/EDA combines global information that is extracted by the EDA with differential information that is obtained by DE to create promising solutions and explore the search space more effectively. In the DE/EDA offspring-generation scheme, one part of a new solution is generated in the DE way, while the other part is sampled from a probability model based on elite solutions. In such a way, both global information and local information are used to guide the further search.

### C. Differences and Connections With Previous Taxonomies

There exist several taxonomies in the literature, which present certain classification of hybrid metaheurisitcs [15], [18], [146], EAs [60], or MAs [19]. Talbi proposed a taxonomy for hybrid metaheuristics, which considers solutions to design and implementation issues [15]. Regarding each issue, Talbi adopted both a hierarchical classification and a flat one. Jourdan *et al*. extended Talbi's taxonomy to the case of cooperation between metaheuristics and exact methods [146]. The taxonomy that is proposed by Raidl [18] is very similar to Talbi's taxonomy, but it removes some flat classification mechanisms and introduces a new classification mechanism, i.e., control strategy, to differentiate integrative hybridization and collaborative hybridization. In integrative hybridization, one parent is considered as a subordinate, embedded component of another parent. In collaborative hybridization, parents exchange information, but are not part of each other. The collaborative hybridization and the integrative hybridization reflect the collaboration-based PR and the embedding-based PR, respectively. Krasnogor and Smith proposed a scheduler-based taxonomy for the classification of different MAs [19]. The MA taxonomy provides a very useful tool to capture the interaction between LS and standard evolutionary operators (mutation, crossover, selection). Calégari *et al*. proposed a taxonomy of EAs, namely table of EAs (TEA), to distinguish different classes of EAs by the enumeration of the fundamental ingredients of each of these algorithms [60]. The TEA is competent for the identification of the differences between EAs in detail; however, it does not provide any classification mechanisms of hybridization strategies. In the following, the new taxonomy that we proposed will be compared with Talbi's taxonomy.

Talbi introduced many classification criteria in his taxonomy, including "low level versus high level," "relay versus teamwork," "homogeneous versus heterogeneous," "global versus partial," "specialist versus general," etc. Distinct from the notions of HL and OL, the low/high level that Talbi mentioned refers to the issue of whether a given function of a parent optimizer is replaced by another one (low level: replaced; high level: self-contained). Relay is *de facto* equivalent to the *sequential* OO. In contrast, teamwork hybridization represents "cooperative optimization models, which involve many parallel cooperating agents, where each agent carries out a search in a solution space" [15]. Therefore, the term "teamwork" does not refer to the *parallel* OO of parents but highlights the feature of cooperatively improving multiple solutions. The homogeneity here (homogeneous/heterogeneous) indicates whether the parent optimizers are same or not. A global hybrid means that all parent optimizers work in the whole search space. In contrast, parents of a partial hybrid are designated to different subproblems, which are determined by a decomposition of the original problem. Obviously, partial hybrids can be categorized into the class ⟨C,C,?,?⟩. The criterion "specialist/general" indicates whether or not some parent optimizers are used to solve other problems beyond searching in solution space (general: no; specialist: yes). In particular, a hybrid in which a parent is used to optimize another one can be classified as a specialist hybrid. Therefore, the specialist hybridization conceptually shares much similarity with the assistance-based hybridization. The OL of parents, the TIT, and the TTI are not considered in Talbi's taxonomy.

According to Talbi's taxonomy, the well-known MA will be placed within the low-level teamwork class [15], [19]. In contrast, the new taxonomy indicates that the MA is a hybridization framework with diversified implementation schemes. An MA can be an instance of ⟨C,?,?,?⟩, where the mapping (operations) of global search and LS can be separated, and for example, LS can be implemented in each generation after a GA completes all of its operations. In this case, the mapping of the MA can be denoted by $(\mathrm{GA} : f_S \circ f_C \circ f_M) \circ f_{\mathrm{LS}}$, where $f_S$, $f_C$, $f_M$, and $f_{\mathrm{LS}}$ represent the mapping of selection, crossover, mutation, and LS operators, respectively. The GA's original consecutive operation can be extracted from the composite mapping and separated from LS. An MA can also be embedding based, where the

mappings of global search and LS cannot be separated. For example, if LS is implemented after both crossover and mutation of the GA [147], the mapping of LS will be integrated (inserted) into the GA, that is to say, the GA's mapping in this case cannot be extracted from the MA as $f_S \circ f_C \circ f_M$. The mapping of this MA can be denoted by $\underline{f_S \circ (f_C} \circ f_{\mathrm{LS}}) \circ (\underline{f_M} \circ f_{\mathrm{LS}})$. It is obvious that the original consecutive operation of the GA, which is denoted by $f_S \circ f_C \circ f_M$, is broken by LS. In addition, even within ⟨C,?,?,?⟩, an MA can be achieved by different schemes, such as ⟨C,P,S,S⟩ (e.g., GQD [21]), ⟨C,P,S,D⟩ (e.g., MOGLS [22]), and ⟨C,I,S,D⟩ (e.g., MA-S1 and MA-S2 [24]).

By comparison, the five hybridization factors in the new taxonomy that is proposed in this paper give rise to a *more comprehensive*, *coherent*, *hierarchical* classification tool, benefiting a step-by-step analysis/design of any hybrid.

## V. Discussion and Future Research on Hybrid Optimization

As can be seen from the proposed taxonomy, designers have manifold choices to design a hybrid optimizer. Here, we have no preferential weighting of any hybridization scheme and try to keep an open mind on different hybridization strategies. Besides, we have no intention to identify the performance differences of various hybridization strategies through some benchmark tests, since an incomplete comparison may mislead other researchers into a prejudice against some hybridization strategies. After all, a successful implementation of certain hybridization strategy is related to many design issues, such as the selection of parent optimizers and the aforementioned design details. Exactly as is claimed at the outset, the review paper aims at to unfold a broader and systematic view of hybridization strategies, as well as to promote the development of more efficient hybrid optimizers. In the following, we will discuss a pivotal issue regarding the design of hybrid optimizers, i.e., the Tr:Er&Ei, as well as point out some research lines about hybrid optimizers.

### A. Tradeoff Between Exploration and Exploitation

Exploration and exploitation are also referred to as diversification (breadth first) and intensification (depth first), respectively [16], [148]. In [148], through an extensive investigation of hybrid metaheuristics that are based on EAs, Lozano and Martínez concluded that the use of EAs specializing in diversification and intensification (Er&Ei) to build hybrid metaheuristics becomes a prospective line of research to obtain effective search algorithms.

Admittedly, the Tr:Er&Ei is a core of all kinds of optimizers, which can be seen from the ideas of the inventors of different hybrids. Generally, excessive exploitation will depress diversity and induce premature convergence. Excessive exploration will result in slow convergence. Therefore, an appropriate Tr:Er&Ei is essential for efficient problem solving. Chen *et al*. pointed out that the best Tr:Er&Ei is dominated by the optimization hardness of problems to be optimized, and it will gradually lean toward exploration as optimization hardness increases [149]. The diversity of practical problems gives rise to diverse demands on the Tr:Er&Ei. In essence, the hybridization of optimizers is a means of adjusting the Tr:Er&Ei. Usually, various optimizers differ in their exploration and exploitation patterns, as well as the Tr:Er&Ei. For example, an important motivation behind many hybrids, such as MAs, is the synergy of explorative (global) search and exploitative (local) search. However, the rationale in terms of the Tr:Er&Ei is not very clear in embedding-based hybridization and assistance-based hybridization. It is hard to predict the performance of an embedding-based hybrid according to the performance of its parents, since the hybridization is neither a temporal assignment nor a spatial assignment of the parents in search space, but an integration that gives birth to a new holistic optimizer. In this sense, embedding-based hybridization usually implies more *empirical* attempts and a *higher risk* of design failure. As for assistance-based hybrids, the assistant parent is expected to improve the search behavior of the master parent, and its contribution to performance improvement has to be attached to the framework of its master, consequently being hard to estimate. Therefore, the Tr:Er&Ei in assistance-based hybrids essentially relies much on a delicate regulation of the Tr:Er&Ei about the master parent. Note that the statement given earlier is not negating the embedding PR or assistance PR, but reminding designers of the two hybridization schemes' basic features.

In most cases, incorporation of LSs into explorative optimizers is a promising, feasible, and efficient hybridization pattern to draw a better Tr:Er&Ei, especially to solve complex combinatorial optimization problems. This is one of the important reasons for the success of many efficient hybrid optimizers, especially those that are frequently cited and whose high quality and efficiency have been proved in practice [20]–[27]. Like other population-based global optimizers, DE and PSO stress on exploration and lack exploitation; therefore, many researchers have tried to hybridize them with LS to strike a balance [67]–[71], [150]–[153]. However, how to incorporate LSs to ensure the expected problem-solving efficiency is still an open issue. See the MA tutorial of Krasnogor and Smith for more design issues on how to balance global search and LS [19].

Although we focus in this paper on the hybridization of two parent optimizers, the proposed taxonomy can also be utilized to analyze some intricate hybridization that involves more parents. One can also follow the hierarchical framework that is shown in Fig. 4 to determine primary hybridization factors one by one to design a hybrid that involves more parent optimizers. When designing a hybrid, one should also consider the *computation complexity* that is possibly introduced by the incorporation of too many optimizers. However, a sophisticated hybridization that is based on multiple parents may be prospective for time-consuming high-dimensional numerical optimization, since the computational cost that is introduced by complicated hybridization is usually insignificant in contrast to the inherent computation complexity and optimization hardness of high-dimensional problems.

### B. Some Topics Worthy of Future Research

*1) ⟨C,C,?,?⟩ Needs to be more explored:* According to the analysis of existing DEPSOs and other representa-

tive hybrids, the hybridization patterns ⟨C,P,?,?⟩ (especially, ⟨C,P,S,D⟩), ⟨C,S,?,?⟩ (especially, ⟨C,S,P,D⟩), ⟨C,I,?,?⟩ (especially, ⟨C,I,S,D⟩) have attracted more attention of researchers. It is expected that more efficient hybrids utilizing the pattern ⟨C,C,?,?⟩ will emerge in the future. For example, the component-level adaptive or self-adaptive alternation of parents, which are mentioned in Section IV-A (see Fig. 2), is a promising research line as it enriches the diversity of evolutionary operations to a large extent. Besides, the cooperative coevolution (CC) that is often employed for high-dimensional numerical optimization is an instance of the hybridization strategy ⟨C,C,S,D⟩ [46], [154]. However, it has merely been successfully implemented on homogeneous optimizers, such as multiple PSOs [46], [154] and multiple DEs [155]. It is possible that the CC's efficiency can be enhanced by the hybridization of diverse optimizers that have different features in exploration–exploitation patterns and the Tr:Er&Ei. In addition, the hybridization of global search (especially, POs) and LS methods at *component* level is also a very interesting topic for future research.

*2) Exploration Pattern in Memetic Algorithms Can be Enriched:* Although the concept of MAs has been extended with its global search being achieved by any explorative PO [156], the current focus of MAs is on the coevolution (adaptation) of different LS methods (LSMs), i.e., individual learning methods [19], [24], [133], [134], [156], [157], which takes into account the diversity of LSMs and the improvement in exploitation. However, the adaptation of global search methods (GSMs) within the MA framework, i.e., the coevolution of multiple GSMs, is seldom considered, which may limit the exploration capability of MAs. A sophisticated tuning of the Tr:Er&Ei in MAs is still worth investigation. In fact, MAs usually represent a paradigm of achieving a better Tr:Er&Ei by combining GSMs and LSMs in a deterministic alternation, though some recent MA researches have provided new schemes to balance evolution (global search) and individual learning (LS) [156], [157].

*3) Typical Hybridization Frameworks Beyond Memetic Algorithms:* In contrast to MAs, parent optimizers within some different hybridization frameworks [108], [158], [159], regardless of their explorative or exploitative nature, will be chosen through competition or switch adaptively, running in the form of adaptive alternation. In particular, the hyperheuristics [160], [161], which is also termed as superheuristics [162], represent a hierarchical hybridization framework, which employs a high-level methodology (e.g., machine-learning techniques and evolutionary algorithms) to schedule a set of low-level heuristics. Different low-level heuristics will be applied at any given time, depending on the current problem state or search stage [161]. An important motivation behind hyperheuristics is to devise an advanced algorithm (system), which can handle classes of problems rather than to solve only one problem, by combining the strength and compensating for the weakness of known heuristics [161], [163]. The choice of low-level heuristics can also be encoded and evolved in a self-adaptive manner (see Fig. 2). Obviously, hybridization in hyperheuristics involves both assistance and collaboration. The high-level methodology serves as an assistant/scheduler of all low-level heuristics, while these heuristics collaborate with each other in search space. It is expected that a sophisticated ensemble based on the hyperheuristic framework can be built to hybridize and coordinate different explorative or/and exploitative optimizers for a wide range of problem solving.

*4) Adapting the Tr:Ei&Er to Problems:* The well-known *No Free Lunch* (NFL) theorem states that no optimizer could keep optimal over all problems, alluding to a compromise between the generality and specialty of an optimizer [164]. However, the following question overriding the NFL theorem may hopefully lead to some powerful hybrid optimizers that are capable of handling a wide range of problems efficiently:

> How to choose the most suitable optimizer in different cases to solve the problem in hand or specific problem instances?

No doubt that the choice of the "best" optimizer is problem dependent, and even instance dependent. Therefore, the adaptation of the Tr:Er&Ei of hybrids to the features of *specific* problems, such as parameter combinations and optimization hardness, by the utilization of *problem-specific* knowledge beforehand or in search process is also a challenging research line worthy of devotion.

## VI. Conclusion

Hybridization has turned out to be an effective and efficient way to design high-performance optimizers, which is witnessed by the rapid evolution of diverse hybrid optimizers in the past decade. As a special and representative member in the family of hybrid optimizers, DEPSO has received much attention from researchers that are interested in optimization, problem solving, and algorithm design. Based on a comprehensive survey on existing DEPSOs, this paper builds a systematic hierarchical taxonomy of hybridization strategies, providing both a tool to analyze hybridization strategies and a reference for designing new optimizers. The proposed taxonomy was utilized to make an analysis of the status quo of DEPSO research, indicating that there is still a major part of hybridization strategies that is involved in the taxonomy, which has not been explored yet. At present, three collaboration-based hybridization strategies, namely ⟨C,P,S,D⟩, ⟨C,S,P,D⟩, and ⟨C,I,S,D⟩, largely dominate the existing DEPSOs. The taxonomy of hybridization strategies, though built on the basis of DEPSO, can be applied to analyze a large spectrum of hybrid optimizers. Finally, this paper presents a discussion on a crucial issue in hybridization design, i.e., the Tr:Er&Ei, and unfolds some promising research directions worthy of future devotion.

## Acknowledgment

The authors would like to thank the Editor-in-Chief, the Associate Editor, and anonymous reviewers for their constructive suggestions to improve the quality of this paper. They would also like to thank the two authors of [39], Prof. Z. F. Hao and Mr. G. H. Guo, who provided very useful materials about their hybrid algorithm.

## REFERENCES


[1] G. B. Dantzig, *Linear Programming and Extensions*. Reading, NJ: Princeton University Press, 1963.

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

[3] H. P. Schwefel, "Evolution strategies: A family of nonlinear optimization techniques based on imitating some principles of organic evolution," *Ann. Oper. Res.*, vol. 1, no. 2, pp. 165–167, 1984.

[4] N. Hansen and A. Ostermeier, "Adapting arbitrary normal mutation distribution in evolution strategies: The covariance matrix adaptation," in *Proc. IEEE Conf. Evol. Comput.*, Nagoya, Japan, May 1996, pp. 312–317.

[5] X. Yao, Y. Liu, and G. Lin, "Evolutionary programming made faster," *IEEE Trans. Evol. Comput.*, vol. 3, no. 2, pp. 82–102, Jul. 1999.

[6] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[7] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE Comput. Intell. Mag.*, vol. 1, no. 4, pp. 28–39, Nov. 2006.

[8] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *Proc. IEEE Int. Conf. Neural Netw.*, Perth, WA, Nov./Dec. 1995, pp. 1942–1948.

[9] R. Storn and K. Price, "Differential evolution—A simple and efficient heuristics for global optimization over continuous spaces," *J. Global Optim.*, vol. 11, no. 4, pp. 341–359, 1997.

[10] P. Larrañaga and J. A. Lozano, *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*. Boston, MA: Kluwer, 2002.

[11] C. Blum, M. J. Blesa Aguilera, A. Roli, and M. Sampels, Eds., *Hybrid Metaheuristics: An Emerging Approach To Optimization* (Studies in Computational Intelligence Series 114). Berlin: Springer-Verlag, 2008.

[12] S. Voß. (2006). "Hybridizing metaheuristics: The road to success in problem solving," in *6th Eur. Conf. Evol. Comput. Combinat. Optim.* (Slides of an invited talk at the EvoCOP 2006), Budapest, Hungary. [Online]. Available: http://www.ads.tuwien.ac.at/evocop/Media:Invited-talk-EvoCOP2006-voss.pdf

[13] M. J. Blesa Aguilera, C. Blum, C. Cotta, A. J. Fernández Leiva, J. E. Gallardo Ruiz, A. Roli, and M. Sampels, Eds., *Proceedings of the. 5th International Workshop on Hybrid Metaheuristics* (Lecture Notes in Computer Science Series 5296). Berlin: Springer-Verlag, 2008.

[14] M. J. Blesa Aguilera, C. Blum, L. Gaspero, A. Roli, M. Sampels, and A. Schaerf, Eds., *Proceedings of the 6th International Workshop on Hybrid Metaheuristics* (Lecture Notes in Computer Science Series 5818). Berlin: Springer-Verlag, 2009.

[15] E. G. Talbi, "A taxonomy of hybrid metaheuristics," *J. Heur.*, vol. 8, no. 5, pp. 541–564, 2002.

[16] M. Gendreau and J. Y. Potvin, "Metaheuristics in combinatorial optimization," *Ann. Oper. Res.*, vol. 140, no. 1, pp. 189–213, 2005.

[17] C. Blum and A. Roli, "Metaheuristics in combinatorial optimization: Overview and conceptual comparison," *ACM Comput. Surv.*, vol. 35, no. 3, pp. 268–308, 2003.

[18] G. R. Raidl, "A unified view on hybrid metaheuristics," in *Proc 3rd Int. Workshop Hybrid Metaheuristics*, Gran Canaria, Spain, Oct. 2006, pp. 1–12.

[19] N. Krasnogor and J. Smith, "A tutorial for competent memetic algorithms: Model, taxonomy, and design issues," *IEEE Trans. Evol. Comput.*, vol. 9, no. 5, pp. 474–488, Oct. 2005.

[20] C. F. Juang, "A hybrid of genetic algorithm and particle swarm optimization for recurrent network design," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 34, no. 2, pp. 997–1006, Apr. 2004.

[21] J. M. Renders and S. P. Flasse, "Hybrid methods using genetic algorithms for global optimization," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 26, no. 2, pp. 243–258, Apr. 1996.

[22] H. Ishibuchi and T. Murata, "A multi-objective genetic local search algorithm and its application to flowshop scheduling," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 28, no. 3, pp. 392–403, Aug. 1998.

[23] B. Liu, L. Wang, Y. H. Jin, F. Tang, and D. X. Huang, "Improved particle swarm optimization combined with chaos," *Chaos Soliton. Fract.*, vol. 25, no. 5, pp. 1261–1271, 2005.

[24] Y. S. Ong and A. J. Keane, "Meta-Lamarckian learning in memetic algorithms," *IEEE Trans. Evol. Comput.*, vol. 8, no. 2, pp. 99–110, Apr. 2004.

[25] L. Wang and D. Z. Zheng, "An effective hybrid optimization strategy for job-shop scheduling problems," *Comput. Oper. Res.*, vol. 28, no. 6, pp. 585–596, 2001.

[26] J. Yen, J. C. Liao, B. J. Lee, and D. Randolph, "A hybrid approach to modeling metabolic systems using a genetic algorithm and simplex method," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 28, no. 2, pp. 173–191, Apr. 1998.

[27] J. Y. Sun, Q. F. Zhang, and E. P. K. Tsang, "DE/EDA: A new evolutionary algorithm for global optimization," *Inf. Sci.*, vol. 169, no. 3–4, pp. 249–262, 2005.

[28] W. J. Zhang and X. F. Xie, "DEPSO: Hybrid particle swarm with differential evolution operator," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Washington, DC, Oct. 2003, pp. 3816–3821.

[29] Y. S. Ong, N. Krasnogor, and H. Ishibuchi, "Special issue on memetic algorithms," *IEEE Trans. Syst. Man Cybern. B, Cybern.*, vol. 37, no. 1, pp. 2–5, Feb. 2007.

[30] K. Price, R. M. Storn, and J. A. Lampinen, *Differential Evolution: A Practical Approach to Global Optimization* (Natural Computing Series), 1st ed. New York: Springer-Verlag, 2005.

[31] U. K. Chakraborty, Ed., *Advances in Differential Evolution*. Berlin: Springer-Verlag, 2008.

[32] S. Das and P. N. Suganthan, "Differential evolution: A survey of the state of the art," *IEEE Tran. Evol. Comput.*, vol. 5, no. 1, pp. 4–31, Feb. 2011.

[33] S. Das, A. Abraham, and A. Konar, "Automatic clustering using an improved differential evolution algorithm," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 38, no. 1, pp. 218–237, Jan. 2008.

[34] C. M. Kwan and C. S. Chang, "Timetable synchronization of mass rapid transit system using multiobjective evolutionary approach," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 38, no. 5, pp. 636–648, Sep. 2008.

[35] C. H. Chen, C. J. Lin, and C. T. Lin, "Nonlinear system control using adaptive neural fuzzy networks based on a modified differential evolution," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 39, no. 4, pp. 459–473, Jul. 2009.

[36] J. Brest, S. Greiner, B. Bošković, M. Mernik, and V. Žumer, "Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems," *IEEE Trans. Evol. Comput.*, vol. 10, no. 6, pp. 646–657, Dec. 2006.

[37] A. K. Qin, V. L. Huang, and P. N. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization," *IEEE Trans. Evol. Comput.*, vol. 13, no. 2, pp. 398–417, Apr. 2009.

[38] K. V. Price, "Differential evolution vs. the functions of the 2nd ICEO," in *Proc. IEEE Int. Conf. Evol. Comput.*, Indianapolis, IN, Apr. 1997, pp. 153–157.

[39] Z. F. Hao, G. H. Guo, and H. Huang, "A particle swarm optimization algorithm with differential evolution," in *Proc. 6th Int. Conf. Mach. Learn. Cybern.*, Hong Kong, China, Aug. 2007, pp. 1031–1035.

[40] J. Kennedy, R. C. Eberhart, and Y. Shi, *Swarm Intelligence*. San Francisco, CA: Morgan Kaufmann, 2001.

[41] Y. del Valle, G. K. Venayagamoorthy, S. Mohagheghi, J. C. Hernandez, and R. G. Harley, "Particle swarm optimization: Basic concepts, variants and applications in power systems," *IEEE Trans. Evol. Comput.*, vol. 12, no. 2, pp. 171–195, Apr. 2008.

[42] P. Kanakasabapathy and K. S. Swarup, "Evolutionary tristate PSO for strategic bidding of pumped-storage hydroelectric plant," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 40, no. 4, pp. 460–471, Jul. 2010.

[43] C. J. Lin, C. H. Chen, and C. T. Lin, "A hybrid of cooperative particle swarm optimization and cultural algorithm for neural fuzzy networks and its prediction applications," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 39, no. 1, pp. 55–68, Jan. 2009.

[44] R. C. Eberhart and Y. Shi, "Comparing inertia weights and constriction factors in particle swarm optimization," in *Proc. IEEE Cong. Evol. Comput.*, La Jolla, CA, Jul. 2000, pp. 84–88.

[45] M. Clerc and J. Kennedy, "The particle swarm: Explosion, stability and convergence in a multidimensional complex space," *IEEE Trans. Evol. Comput.*, vol. 6, no. 1, pp. 58–73, Feb. 2002.

[46] F. van den Bergh and A. P. Engelbrecht, "A cooperative approach to particle swarm optimization," *IEEE Trans. Evol. Comput.*, vol. 8, no. 3, pp. 225–239, Jun. 2004.

[47] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions," *IEEE Trans. Evol. Comput.*, vol. 10, no. 3, pp. 281–295, Jun. 2006.

[48] J. Kennedy, "Small world and mega-minds: Effects of neighborhood topology on particle swarm performance," in *Proc. IEEE Cong. Evol. Comput.*, Washington, DC, Jul. 1999, pp. 1931–1938.

[49] J. Kennedy and R. Mendes, "Population structure and particle swarm performance," in *Proc. IEEE Cong. Evol. Comput.*, Honolulu, HI, May 2002, pp. 1671–1676.

[50] J. Kennedy and R. Mendes, "Neighborhood topologies in fully informed and best-of-neighborhood particle swarms," *IEEE Trans. Syst. Man Cybern. C, Appl. Rev.*, vol. 36, no. 4, pp. 515–519, Jul. 2006.

[51] J. Sun, Q. C. Zhao, and P. B. Luh, "A unified optimization framework for population-based methods," in *Proc. 4th IEEE Int. Conf. Autom. Sci. Eng.*, Arlington, VA, Aug. 2008, pp. 383–387.

[52] M. M. Ali and L. P. Fatti, "A differential free point generation scheme in the differential evolution algorithm," *J. Global Optim.*, vol. 35, no. 4, pp. 551–572, 2006.

[53] R. Poli, "Mean and variance of the sampling distribution of particle swarm optimizers during stagnation," *IEEE Trans. Evol. Comput.*, vol. 13, no. 4, pp. 712–721, Aug. 2009.

[54] J. Kennedy, "Bare bones particle swarms," in *Proc. IEEE Swarm Intell. Symp.*, Indianapolis, IN, Apr. 2003, pp. 80–87.

[55] J. Kennedy, "The particle swarm as collaborative sampling of the search space," *Adv. Complex Syst.*, vol. 10, no. 1, pp. 191–213, 2007.

[56] M. G. H. Omran, A. P. Engelbrecht, and A. Salman, "Self-adaptive barebones differential evolution," in *Proc. IEEE Cong. Evol. Comput.*, Singapore, Sep. 2007, pp. 2858–2865.

[57] X. F. Xie and W. J. Zhang, "SWAF: Swarm algorithm framework for numerical optimization," in *Proc. 6th Ann. Conf. Genetic Evol. Comput.*, Seattle, WA, Jun. 2004, pp. 238–250.

[58] É. D. Taillard, L. M. Gambardella, M. Gendreau, and J. Y. Potvin, "Adaptive memory programming: a unified view of metaheuristics," *Eur. J. Oper. Res.*, vol. 135, no. 1, pp. 1–16, 2001.

[59] F. Glover, "Tabu search—Part I," *ORSA J. Comput.*, vol. 1, no. 3, pp. 190–206, 1989.

[60] P. Calégari, G. Coray, A. Hertz, D. Kobler, and P. Kuonen, "A taxonomy of evolutionary algorithms in combinatorial optimization," *J. Heur.*, vol. 5, no. 2, pp. 145–158, 1999.

[61] J. Teo, "Exploring dynamic self-adaptive populations in differential evolution," *Soft Comput. A, Fusion Found. Method. Appl.*, vol. 10, no. 8, pp. 673–686, 2006.

[62] N. S. Teng, J. Teo, and M. H. A. Hijazi, "Self-adaptive population sizing for a tune-free differential evolution," *Soft Comput. A, Fusion Found. Methodol. Appl.*, vol. 13, no. 7, pp. 709–724, 2009.

[63] J. Brest and M. S. Maučec, "Population size reduction for the differential evolution algorithm," *Appl. Intell.*, vol. 29, no. 3, pp. 228–247, 2008.

[64] S. T. Hsieh, T. Y. Sun, C. C. Liu, and S. J. Tsai, "Efficient population utilization strategy for particle swarm optimizer," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 39, no. 2, pp. 444–456, Apr. 2009.

[65] W. F. Leong and G. G. Yen, "PSO-based multiobjective optimization with dynamic population size and adaptive local archives," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 38, no. 5, pp. 1270–1293, Oct. 2008.

[66] P. Greistorfer and S. Voß, "Controlled pool maintenance for metaheuristics," in *Metaheuristic Optimization via Memory and Evolution: Tabu Search and Scatter Search* (Operations Research/Computer Science Interfaces Series 30), R. Sharda, S. Voß, C. Rego, and B. Alidaee, Eds. Berlin: Springer-Verlag, 2005, part V, pp. 387–424.

[67] N. Noman and H. Iba, "Accelerating differential evolution using an adaptive local search," *IEEE Trans. Evol. Comput.*, vol. 12, no. 1, pp. 107–125, Feb. 2008.

[68] B. Qian, L. Wang, D. X. Huang, W. L. Wang, and X. Wang, "An effective hybrid DE-based algorithm for multi-objective flow shop scheduling with limited buffers," *Comput. Oper. Res.*, vol. 36, no. 1, pp. 209–233, 2009.

[69] F. Neri and E. Mininno, "Memetic compact differential evolution for Cartesian robot control," *IEEE Comput. Intell. Mag.*, vol. 5, no. 2, pp. 54–65, May 2010.

[70] B. Liu, L. Wang, and Y. H. Jin, "An effective PSO-based memetic algorithm for flow shop scheduling," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 37, no. 1, pp. 18–27, Feb. 2007.

[71] B. B. Li, L. Wang, and B. Liu, "An effective PSO-based hybrid algorithm for multiobjective permutation flow shop scheduling," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 38, no. 4, pp. 818–831, Jul. 2008.

[72] T. Hendtlass, "A combined swarm differential evolution algorithm for optimization problems," *Lecture Notes Comput. Sci.*, vol. 2070, pp. 11–18, 2001.

[73] V. Ramesh, T. Jayabarathi, S. Asthana, S. Mital, and S. Basu, "Combined hybrid differential particle swarm optimization approach for economic dispatch problems," *Elec. Power Compon. Syst.*, vol. 38, no. 5, pp. 545–557, 2010.

[74] H. Talbi and M. Batouche, "Hybrid particle swarm with differential evolution for multimodal image registration," in *Proc. IEEE Conf. Ind. Tech.*, Hammamet, Tunisia, Dec. 2004, pp. 1567–1572.

[75] R. Xu, G. K. Venayagamoorthy, and D. C. WunschII, "Modeling of gene regulatory networks with hybrid differential evolution and particle swarm optimization," *Neural Netw.*, vol. 20, no. 8, pp. 917–927, 2007.

[76] S. Wang, X. Liu, J. Qiu, J. G. Zhu, Y. Guo, and Z. W. Lin, "Robust optimization in HTS cable based on DEPSO and design for six sigma," in *Proc. IEEE Ind. Appl. Soc. Ann. Meeting*, Alberta, Canada, Oct. 2008, pp. 1–5.

[77] B. Luitel and G. K. Venayagamoorthy, "Differential evolution particle swarm optimization for digital filter design," in *Proc. IEEE Cong. Evol. Comput.*, Hong Kong, China, Jun. 2008, pp. 3954–3961.

[78] P. W. Moore and G. K. Venayagamoorthy, "Evolving digital circuit using hybrid particle swarm optimization and differential evolution," *Int. J. Neural Syst.*, vol. 16, no. 3, pp. 163–177, 2006.

[79] B. Liu, L. Wang, Y. H. Jin, and D. X. Huang, "Designing neural networks using hybrid particle swarm optimization," *Lecture Notes Comput. Sci.*, vol. 3496, pp. 391–397, 2005.

[80] H. Huang, Z. H. Wei, Z. Q. Li, and W. B. Rao, "The back analysis of mechanics parameters based on DEPSO algorithm and parallel FEM," in *Proc. Int. Conf. Comput. Intell. Natur. Comput.*, Wuhan, China, Jun. 2009, pp. 81–84.

[81] H. Liu, Z. X. Cai, and Y. Wang, "Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization," *Appl. Soft Comput.*, vol. 10, no. 2, pp. 629–640, 2010.

[82] R. Xu, J. Xu, and D. C. WunschII, "Clustering with differential evolution particle swarm optimization," in *Proc. IEEE Cong. Evol. Comput.*, Barcelona, Spain, Jul. 2010, pp. 1–8.

[83] M. G. Epitropakis, V. P. Plagianakos, and M. N. Vrahatis, "Evolving cognitive and social experience in particle swarm optimization through differential evolution," in *Proc. IEEE Cong. Evol. Comput.*, Barcelona, Spain, Jul. 2010, pp. 1–8.

[84] D. Liu, D. Huang, Y. Tian, and Y. Chen, "Multistage inventory hybrid intelligent optimization under grey fuzzy uncertainty," in *Proc. Int. Conf. Comput. Intell. Secur.*, Guangdong, China, Nov. 2006, pp. 514–519.

[85] W. Xu and X. S. Gu, "A hybrid particle swarm optimization approach with prior crossover differential evolution," in *Proc. 1st ACM/SIGEVO Summit Genetic Evol. Comput.*, Shanghai, China, Jun. 2009, pp. 671–677.

[86] D. F. Ning, W. G. Zhang, and B. Li, "Differential evolution based particle swarm optimizer for neural network learning," in *Proc. 7th World Cong. Intell. Contr. Autom.*, Chongqing, China, Jun. 2008, pp. 4444–4447.

[87] A. Caponio, F. Neri, and V. Tirronen, "Super-fit control adaptation in memetic differential evolution frameworks," *Soft Comput. A, Fusion Found. Method. Appl.*, vol. 13, no. 8–9, pp. 811–831, 2009.

[88] M. Ali, M. Pant, and A. Abraham, "Inserting information sharing mechanism of PSO to improve the convergence of DE," in *Proc. World Cong. Natur. Bio. Inspir. Comput.*, Coimbatore, India, Dec. 2009, pp. 282–287.

[89] D. W. Gong, Y. Zhang, and C. L. Qi, "Environmental/economic power dispatch using a hybrid multi-objective optimization algorithm," *Int. J. Elec. Power Energy Syst.*, vol. 32, no. 6, pp. 607–614, 2010.

[90] J. Grobler and A. P. Engelbrecht, "Hybridizing PSO and DE for improved vector evaluated multi-objective optimization," in *Proc. IEEE Cong. Evol. Comput.*, Trondheim, Norway, May 2009, pp. 1255–1262.

[91] S. W. Jiang and Z. H. Cai, "A novel hybrid particle swarm optimization for multi-objective problems," *Lecture Notes Comput. Sci.*, vol. 5855, pp. 28–37, 2009.

[92] B. Niu and L. Li, "A novel PSO-DE-based hybrid algorithm for global optimization," in *Proc. 4th Int. Conf. Intell. Comput.*, Shanghai, China, Sep. 2008, pp. 156–163.

[93] B. Niu and L. Li, "Design of T–S fuzzy model based on PSODE algorithm," in *Proc. 4th Int. Conf. Intell. Comput.*, Shanghai, China, Sep. 2008, pp. 384–390.

[94] M. Zhang, W. Zhang, and Y. Sun, "Chaotic co-evolutionary algorithm based on differential evolution and particle swarm optimization," in *Proc. IEEE Conf. Autom. Logist.*, Shenyang, China, Aug. 2009, pp. 885–889.

[95] X. Wang, Q. Yang, and Y. Zhao, "Research on hybrid PSODE with triple populations based on multiple differential evolutionary models," in *Proc. Int. Conf. Elec. Contr. Eng.*, Wuhan, China, Jun. 2010, pp. 1692–1696.

[96] M. R. Sentinella and L. Casalino, "Hybrid evolutionary algorithm for the optimization of interplanetary trajectories," *J. Spacecraft Rockets*, vol. 46, no. 2, pp. 365–372, 2009.

[97] M. R. Sentinella and L. Casalino, "Cooperative evolutionary algorithm for space trajectory optimization," *Celest. Mech. Dyn. Astr.*, vol. 105, no. 1–3, pp. 211–227, 2009.

[98] K. Yang, K. Maginu, and H. Nomura, "Cultural algorithm-based quantum-behaved particle swarm optimization," *Int. J. Comput. Math.*, vol. 87, no. 10, pp. 2143–2157, 2010.

[99] F. Peng, K. Tang, G. L. Chen, and X. Yao, "Population-based algorithm portfolios for numerical optimization," *IEEE Trans. Evol. Comput.*, vol. 14, no. 5, pp. 782–800, Oct. 2010.

[100] F. Hu and F. Wu, "Diploid hybrid particle swarm optimization with differential evolution for open vehicle routing problem," in *Proc. 8th World Cong. Intell. Contr. Autom.*, Jinan, China, Jul. 2010, pp. 2692–2697.

[101] Y. Wu, X. Z. Gao, X. L. Huang, and K. Zenger, "A hybrid optimization method of particle swarm optimization and cultural algorithm," in *Proc. 6th Int. Conf. Natur. Comput.*, Yantai, China, Aug. 2010, pp. 2515–2519.

[102] D. De, S. Ray, A. Konar, and A. Chatterjee, "An evolutionary SPDE breeding-based hybrid particle swarm optimizer: application in coordination of robot ants for camera coverage area optimization," in *Proc. 1st Int. Conf. Pattern Recog. Mach. Intell.*, Kolkata, India, Dec. 2005, pp. 413–416.

[103] M. G. H. Omran, A. P. Engelbrecht, and A. Salman, "Differential evolution based particle swarm optimization," in *Proc. IEEE Swarm Intell. Symp.*, Honolulu, HI, Apr. 2007, pp. 112–119.

[104] P. Kim and J. Lee, "An integrated method of particle swarm optimization and differential evolution," *J. Mech. Sci. Tech.*, vol. 23, no. 2, pp. 426–434, 2009.

[105] H. Dhahri, A. M. Alimi, and F. Karray, "The modified particle swarm optimization for the design of the beta basis function neural networks," in *Proc. IEEE Cong. Evol. Comput.*, Hong Kong, China, Jun. 2008, pp. 3874–3880.

[106] M. Pant, R. Thangaraj, C. Grosan, and A. Abraham, "Hybrid differential evolution: Particle swarm optimization algorithm for solving global optimization problems," in *Proc. 3rd Int. Conf. Digit. Info. Manage.*, London, U.K, Nov. 2008, pp. 18–24.

[107] S. Khamsawang, P. Wannakarn, and S. Jiriwibhakorn, "Hybrid PSO-DE for solving the economic dispatch problem with generator constraints," in *Proc. 2nd Int. Conf. Comput. Autom. Eng.*, Singapore, Feb. 2010, pp. 135–139.

[108] K. Takano and M. Hagiwara, "An integrated framework of hybrid evolutionary computations," in *Proc. IEEE Cong. Evol. Comput.*, Trondheim, Norway, May 2009, pp. 838–845.

[109] B. Xin, J. Chen, Z. H. Peng, and F. Pan, "An adaptive hybrid optimizer based on particle swarm and differential evolution for global optimization," *Sci. China Inf. Sci.*, vol. 53, no. 5, pp. 980–989, 2010.

[110] S. Das, A. Konar, and U. K. Chakraborty, "Improving particle swarm optimization with differentially perturbed velocity," in *Proc. Genetic Evol. Comput. Conf.*, Washington, DC, Jun. 2005, pp. 177–184.

[111] X. P. Zhang, W. Ding, J. Y. Wang, Z. S. Fan, and G. F. Deng, "Spatial clustering with obstacles constraints using PSO-DV and K-medoids," in *Proc. 3rd Int. Conf. Intell. Syst. Know. Eng.*, Xiamen, China, Nov. 2008, pp. 246–251.

[112] B. Liu and H. Pan, "A hybrid PSO-DV based intelligent method for fault diagnosis of gear-box," in *Proc. IEEE Int. Symp. Comput. Intell. Robot. Autom.*, Daejeon, Korea, Dec. 2009, pp. 451–456.

[113] J. Li, "A hybrid differential evolution method for practical engineering problems," in *Proc. IITA Int. Conf. Contr. Autom. Syst. Eng.*, Zhangjiajie, China, Jul. 2009, pp. 54–57.

[114] J. Shu and J. Li, "A hybrid of differential evolution and particle swarm optimization for global optimization," in *Proc. 3rd Int. Symp. Intell. Info. Tech. Appl.*, Nanchang, China, Nov. 2009, pp. 138–141.

[115] S. Das, A. Abraham, U. K. Chakraborty, and A. Konar, "Differential evolution using a neighborhood-based mutation operator," *IEEE Trans. Evol. Comput.*, vol. 13, no. 3, pp. 526–553, Jun. 2009.

[116] X. Xu, Y. Li, S. Fang, Y. Wu, and F. Wang, "A novel differential evolution scheme combined with particle swarm intelligence," in *Proc. IEEE Cong. Evol. Comput.*, Hong Kong, China, Jun. 2008, pp. 1057–1062.

[117] S. Kannan, S. M. R. Slochanal, P. Subbaraj, and N. P. Padhy, "Application of particle swarm optimization technique and its variants to generation expansion planning," *Elec. Power Syst. Res.*, vol. 70, no. 3, pp. 203–210, 2004.

[118] S. Jiang, Q. Wang, and J. Jiang, "Particle swarm optimization algorithm based on velocity differential mutation," in *Proc. 21st Chin. Contr. Decision Conf.*, Guilin, China, Jun. 2009, pp. 1860–1865, (in Chinese).

[119] M. G. H. Omran, A. P. Engelbrecht, and A. Salman, "Bare bones differential evolution," *Eur. J. Oper. Res.*, vol. 196, no. 1, pp. 128–139, 2009.

[120] W. R. M. U. K. Wickramasinghe and X. Li, "Choosing leaders for multiobjective PSO algorithms using differential evolution," in *Proc. 7th Int. Conf. Simul. Evol. Learn.*, Melbourne, Australia, Dec. 2008, pp. 249–258.

[121] C. Zhang, J. Ning, S. Lu, D. Ouyang, and T. Ding, "A novel hybrid differential evolution and particle swarm optimization algorithm for unconstrained optimization," *Oper. Res. Lett.*, vol. 37, no. 2, pp. 117–122, 2009.

[122] S. Liu, X. Wang, and X. You, "Cultured differential particle swarm optimization for numerical optimization problems," in *Proc. Int. Conf. Natur. Comput.*, Haikou, China, Aug. 2007, pp. 642–648.

[123] W. Yi, C. Cao, and C. Zhang, "The geometric constraint solving based on hybrid differential evolution and particle swarm optimization algorithm," in *Proc. Int. Conf. Intell. Contr. Info. Process.*, Dalian, China, Aug. 2010, pp. 692–695.

[124] S. Das and S. Sil, "Kernel-induced fuzzy clustering of image pixels with an improved differential evolution algorithm," *Inf. Sci.*, vol. 180, no. 8, pp. 1237–1256, 2010.

[125] D. Liu, Y. Chen, H. Mao, Z. Zhang, and X. Gu, "Optimization of the supply chain production planning programming under hybrid uncertainties," in *Proc. Int. Conf. Intell. Comput. Tech. Autom.*, Hunan, China, Oct. 2008, pp. 1235–1239.

[126] S. Wang, L. Ma, and D. Sun, "Hybrid differential evolution particle swarm optimization algorithm for reactive power optimization," in *Proc. Asia-Pacific Conf. Power Energy Eng.*, Chengdu, China, Mar. 2010, pp. 1–4.

[127] K. Vaisakh, P. Praveena, and S. Rama Mohana Rao, "DEPSO and bacterial foraging optimization based dynamic economic dispatch with nonsmooth fuel cost functions," in *Proc. World Cong. Natur. Bio. Inspir. Comput.*, Coimbatore, India, Dec. 2009, pp. 152–157.

[128] C. Potter, G. K. Venayagamoorthy, and K. Kosbar, "MIMO beamforming with neural network channel prediction trained by a novel PSO-EA-DEPSO algorithm," in *Proc. Int. Joint Conf. Neur. Netw.*, Hong Kong, China, Jun. 2008, pp. 3338–3344.

[129] C. Potter, G. K. Venayagamoorthy, and K. Kosbar, "RNN based MIMO channel prediction," *Signal Process.*, vol. 90, no. 2, pp. 440–450, 2010.

[130] Y. C. Lin, K. S. Hwang, and F. S. Wang, "Co-evolutionary hybrid differential evolution for mixed-integer optimization problems," *Eng. Optim.*, vol. 33, no. 6, pp. 663–682, 2001.

[131] H. A. Abbass, "The self-adaptive Pareto differential evolution algorithm," in *Proc. Cong. Evol. Comput.*, Honolulu, HI, May 2002, pp. 831–836.

[132] M. G. Epitropakis, V. P. Plagianakos, and M. N. Vrahatis, "Balancing the exploration and exploitation capabilities of the differential evolution algorithm," in *Proc. IEEE Cong. Evol. Comput.*, Hong Kong, China, Jun. 2008, pp. 2686–2693.

[133] Y. S. Ong, M. H. Lim, N. Zhu, and K. W. Wong, "Classification of adaptive memetic algorithms: A comparative study," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 36, no. 1, pp. 141–152, Feb. 2006.

[134] J. E. Smith, "Co-evolving memetic algorithms: a review and progress report," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 37, no. 1, pp. 6–17, Feb. 2007.

[135] J. J. Grefenstette, "Optimization of control parameters for genetic algorithms," *IEEE Trans. Syst., Man, Cybern.*, vol. 16, no. 1, pp. 122–128, Jan. 1986.

[136] E. Cantú-Paz, *Efficient and Accurate Parallel Genetic Algorithms*. Norwell, MA: Kluwer, 2000.

[137] I. L. Schoeman and A. P. Engelbrecht, "A novel particle swarm niching technique based on extensive vector operations," *Natur. Comput.*, vol. 9, no. 3, pp. 683–701, 2010.

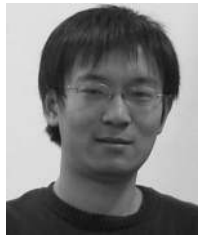
[138] R. Brits, A. P. Engelbrecht, and F. van den Bergh, "Locating multiple optima using particle swarm optimization," *Appl. Math. Comput.*, vol. 189, no. 2, pp. 1859–1883, 2007.

[139] M. Weber, V. Tirronen, and F. Neri, "Scale factor inheritance mechanism in distributed differential evolution,," *Soft Comput. A, Fusion Found. Method. Appl.*, vol. 14, no. 11, pp. 1187–1207, 2010.

[140] X. D. Li, "Efficient differential evolution using speciation for multimodal function optimization,," in *Proc. Genetic Evol. Comput., Conf.*, Washington, DC, Jun. 2005, pp. 873–880.

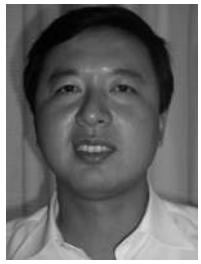
[141] B. Yang, Y. Chen, Z. Zhao, and Q. Han, "A master-slave particle swarm optimization algorithm for solving constrained optimization problems,"

in *Proc. 6th World Cong. Intell. Contr. Autom.*, Dalian, China, Jun. 2006, pp. 3208–3212.
[142] and L. dos R. A. Krohling and S. Coelho, "Coevolutionary particle swarm optimization using Gaussian distribution for solving constrained optimization problems," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 36, no. 6, pp. 1407–1416, Dec. 2006.
[143] B. W. Wah and Y. Chen, "Hybrid constrained simulated annealing and genetic algorithms for nonlinear constrained optimization," in *Proc. IEEE Cong. Evol. Comput.*, May, 2001, pp. 925–932.
[144] S. Das, "Nelder–Mead evolutionary hybrid algorithms," in *Encyclopedia of Artificial Intelligence*, vol. 3, J. R. R. Dopico, J. D. de la Calle, and A. P. Sierra, Eds. New York: IGI Global, 2009, pp. 1191–1196.
[145] S. W. Mahfoud and D. E. Goldberg, "Parallel recombinative simulated annealing: A genetic algorithm," *Parallel Comput.*, vol. 21, no. 1, pp. 1–28, 1995.
[146] L. Jourdan, M. Basseur, and E. G. Talbi, "Hybridizing exact methods and metaheuristics: A taxonomy," *Eur. J. Oper. Res.*, vol. 199, no. 3, pp. 620–629, 2009.
[147] P. Merz and B. Freisleben, "Fitness landscape analysis and memetic algorithms for the quadratic assignment problem," *IEEE Trans. Evol. Comput.*, vol. 4, no. 4, pp. 337–352, Nov. 2000.
[148] M. Lozano and C. García-Martínez, "Hybrid metaheuristics with evolutionary algorithms specializing in intensification and diversification: Overview and progress report," *Comput. Oper. Res.*, vol. 37, no. 3, pp. 481–497, 2010.
[149] J. Chen, B. Xin, Z. Peng, L. Dou, and J. Zhang, "Optimal contraction theorem for exploration–exploitation tradeoff in search and optimization," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 39, no. 3, pp. 680–691, May 2009.
[150] J. Knowles, "ParEGO: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems," *IEEE Trans. Evol. Comput.*, vol. 10, no. 1, pp. 50–66, Feb. 2005.
[151] P. Koduru, S. Das, and S. M. Welch, "Multi-objective hybrid PSO using $\varepsilon$-Fuzzy dominance," in *Proc. Genetic Evol. Comput. Conf.*, London, U.K, Jul. 2007, pp. 853–860.
[152] P. Koduru, Z. Dong, S. Das, S. M. Welch, J. L. Roe, and E. Charbit, "Multi-objective evolutionary-simplex hybrid approach for the optimization of differential equation models of gene networks," *IEEE Trans. Evol. Comput.*, vol. 12, no. 5, pp. 572–590, Oct. 2008.
[153] S. Das, P. Koduru, M. Gui, M. Cochran, A. Wareing, S. M. Welch, and B. R. Babin, "Adding local search to particle swarm optimization," in *Proc. IEEE Cong. Evol. Comput.*, Vancouver, BC, Canada, Jul. 2006, pp. 428–433.
[154] X. D. Li and X. Yao, "Tackling high-dimensional nonseparable optimization problems by cooperatively coevolving particle swarms," in *Proc. IEEE Cong. Evol. Comput.*, Trondheim, Norway, May 2009, pp. 1546–1553.
[155] Z. Yang, K. Tang, and X. Yao, "Large scale evolutionary optimization using cooperative coevolution," *Inf. Sci.*, vol. 178, no. 15, pp. 2985–2999, 2008.
[156] Q. H. Nguyen, Y. S. Ong, and M. H. Lim, "A probabilistic memetic framework," *IEEE Trans. Evol. Comput.*, vol. 13, no. 3, pp. 604–623, Jun. 2009.
[157] Z. Z. Zhou, Y. S. Ong, P. B. Nair, A. J. Keane, and K. Y. Lum, "Combining global and local surrogate models to accelerate evolutionary optimization," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 37, no. 1, pp. 66–76, Jan. 2007.
[158] M. J. Colaco, G. S. Dulikravich, and T. J. Martin, "Control of unsteady solidification via optimized magnetic fields," *Mater. Manuf. Process.*, vol. 20, no. 3, pp. 435–458, 2005.
[159] G. S. Dulikravich, T. J. Martin, B. H. Dennis, and N. F. Foster, "Multidisciplinary hybrid constrained GA optimization," in *EUROGEN'99—Evol. Algorithm Eng. Comput. Sci.: Recent Adv. Ind. App.*, K. Miettinen, M. M. Makela, P. Neittaanmaki, and J. Periaux, Eds. New York, Finland: Wiley, Jyvaskyla, May 1999, pp. 233–259.
[160] E. Burke, E. Hart, G. Kendall, J. Newall, P. Ross, and S. Schulenburg, "Hyper-heuristics: An emerging direction in modern search technology," in *Handbook of Metaheuristics*, F. Glover and G. Kochenberger, Eds. Norwell, MA: Kluwer, 2003, pp. 457–474.
[161] P. Ross, "Hyperheuristics," in *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques*, E. K. Burke and G. Kendall, Eds. Berlin: Springer-Verlag, 2005, pp. 529–556.
[162] D. Corne, K. Deb, J. Knowles, and X. Yao. (2011). "Selected applications of natural computing," in *Handbook of Natural Computing*, G. Rozenberg, T. Bäck, and J. N. Kok, Eds. Berlin: Springer-Verlag. [Online]. Available: http://www.macs.hw.ac.uk/∼dwcorne/Teaching/SANC.pdf
[163] E. Özcan, B. Bilgin, and E. E. Korkmaz, "Hill climbers and mutational heuristics in hyperheuristics," *Lecture Notes Comput. Sci.*, vol. 4193, pp. 202–211, 2006.
[164] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 67–82, Apr. 1997.



**Bin Xin** (S'09–M'10) received the B.S. degree in information engineering from the Beijing Institute of Technology, Beijing, China, in 2004. He is currently working toward the Ph.D. degree at the School of Automation, Beijing Institute of Technology.

His research interests include search and optimization, evolutionary computation, operations research, and combinatorial optimization.



**Jie Chen** (M'09) received the B.S., M.S., and Ph.D. degrees all in control theory and control engineering from the Beijing Institute of Technology, Beijing, China, in 1986, 1993, and 2000, respectively.

He is currently a Professor of Control Science and Engineering, Beijing Institute of Technology. His main research interests include complicated system multiobjective optimization and decision, intelligent control, constrained nonlinear control, and optimization methods.

**Juan Zhang** received the B.S. and Ph.D. degrees in control theory and control engineering from the Beijing Institute of Technology, Beijing, China, in 1997 and 2002, respectively.

She is currently an Associate Professor of Control Science and Engineering, the Beijing Institute of Technology. Her research interests include operations research, intelligent decision makings, distributed simulation, and image processing.

**Hao Fang** received the B.S. degree from the Xi'an University of Technology, Xi'an, China, in 1995, and the M.S. and Ph.D. degrees from the Xi'an Jiaotong University, Xi'an, China, in 1998 and 2002, respectively.

He held two Postdoctoral appointments at the INRIA/France Research Group of COPRIN and at the LASMEA, UNR6602 CNRS/Blaise Pascal University, Clermont-Ferrand, France. Since 2005, he has been an Assistant Professor with the Beijing Institute of Technology, Beijing, China. His research interests include all-terrain mobile robots, robotic control, robotic behavior optimization, and parallel manipulators.

**Zhihong Peng** received the Ph.D. degree in control theory and control engineering from the Central South University, Changsha, China, in 2000.

From December 2000 to February 2003, she was a Postdoctoral Research Associate with Beijing Institute of Technology, Beijing, China. She is currently an Associate Professor with Beijing Institute of Technology. Her current research interests include intelligent information processing and intelligent control.