Expert Systems With Applications 130 (2019) 276–292

Contents lists available at ScienceDirect

Expert Systems With Applications

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

Hierarchical differential evolution algorithm combined with multi-cross operation

Zhao-Guang Liu[a,*], Xiu-Hua Ji[a], Yang Yang[b]

[a] School of Computer Science and Technology, Shandong University of Finance and Economics, Jinan, China
[b] School of Information Science and Engineering, Shandong University, Jinan, China

ARTICLE INFO

Article history:
Received 12 June 2018
Revised 17 April 2019
Accepted 17 April 2019
Available online 18 April 2019

Keywords:
Differential evolution
Particle swarm optimization
Hierarchical structure
Multi-cross operation

ABSTRACT

In expert systems, complex optimization problems are always characterized by nonlinearity, nonconvexity, multi-modality, discontinuity, and high dimensionality. Although classical optimization algorithms are mature, they readily fall into a local optimum. The differential evolution (DE) algorithm has been successfully applied to solve numerous problems with expert systems. However, balancing the global and local search capabilities of the DE algorithm remains an open issue and has attracted significant research attention. Thus, a hierarchical heterogeneous DE algorithm that incorporates multi-cross operation (MCO) is proposed in this article. In the proposed algorithm, success-history-based adaptive DE (SHADE) is implemented in the bottom layer, while MCO is implemented in the top layer. The MCO search is based on the SHADE results, but its search results do not affect the bottom layer. First-order stability analyses conducted for the presented MCO showed that the individual positions are expected to converge at a fixed point in the search space. The accuracy and convergence speed of the proposed algorithm were also experimentally compared with those of eight other advanced particle swarm optimization techniques and DE variants using benchmark functions from CEC2017. The proposed algorithm yielded better solution accuracy for 30- and 50-dimensional problems than the other variants, and although it did not provide the fastest convergence for all of the functions, it ranked among the top three for the unimodal and simple multimodal functions and achieved fast convergence for the other functions.

1. Introduction

In expert systems, numerous real problems can essentially be modeled as optimization tasks, e.g., power systems (Chang, 2010), path planning (Escario, Jimenez, & Giron-Sierra, 2012; Nazarahari, Khanmirza, & Doostie, 2019), product design (Tseng, Chang, Lee, & Huang, 2018), operations management (Wang & Yeh, 2014), and satellite networks (Zhang, Zhang, & Feng, 2014). Since applications of expert systems are very extensive, complex optimization problems are always characterized by nonlinearity, nonconvexity, multimodality, discontinuity, and high dimensionality. Classical optimization algorithms are mature, but they readily fall into a local optimum and are no longer completely effective in solving such complex optimization problems.

Swarm intelligence (SI) is a population-based random search optimization technique that is employed in various algorithms, such as genetic algorithm (GA) (Nazarahari et al., 2019; Tseng et al., 2018), ant colony optimization (ACO) (Chen, Zhou, & Luo, 2017; Liao, Stützle, Ocac, & Dorigo, 2014), particle swarm optimization (PSO) (Lin, Zhang, & Liu, 2018; Marinakis, Migdalas, & Sifaleras, 2017), artificial bee colony (ABC) (Pan, 2016; Zorarpacı & Özel, 2016), and differential evolution (DE) (Mohamed & Suganthan, 2017; Plessis & Engelbrecht, 2012; Sethanan & Neungmatch, 2016) algorithms. As one of the most competitive and versatile SI algorithms, the DE algorithm has been successfully applied to solve numerous real-world problems in expert systems (Coelho & Bernert, 2010; Mlakar, Potocnik, & Brest, 2016; Liu & Sun, 2011; Sethanan & Pitakaso, 2016).

Storn and Price (1995, 1997) proposed the DE algorithm. Each vector in DE represents a possible solution for an optimization problem in a D-dimensional space. After initialization, the DE algorithm enters a loop of evolutionary operations: mutation, crossover, and selection. The iterations continue until a termination criterion (such as exhaustion of the maximum number of functional evaluations) is satisfied. The following notation is adopted to represent the *i*th ($i = 1, 2, \ldots, N$) vector of the population at the

* Correspondence author.
E-mail addresses: liuzhg@sdufe.edu.cn (Z.-G. Liu), jane@sdu.edu.cn (X.-H. Ji), yyang@sdu.edu.cn (Y. Yang).

https://doi.org/10.1016/j.eswa.2019.04.040
0957-4174/

tth ($t = 1, 2, \ldots, T$) iteration:

$$\mathbf{x}_i(t) = \left[x_i^1(t), x_i^2(t), \ldots, x_i^D(t)\right], \quad (1)$$

where d ($d = 1, 2, \ldots, D$) denotes the dimension of the solution space and N is the number of individuals in the population.

To establish a starting point for the optimization process, an initial population must be created. The dth dimension of the ith vector can be obtained as follows:

$$x_i^d(1) = x_{\min}^d + rand(0,1) \cdot \left(x_{\max}^d - x_{\min}^d\right), \quad (2)$$

where $\mathbf{x}_{\min} = [x_{\min}^1, x_{\min}^2, \ldots, x_{\min}^D]$ and $\mathbf{x}_{\max} = [x_{\max}^1, x_{\max}^2, \ldots, x_{\max}^D]$ are the minimum and maximum bounds of the search space, respectively, and $rand(0,1)$ is a uniformly distributed random number within the range of [0, 1] and is instantiated independently for each dimension of the ith vector.

1.1. Mutation

At iteration t, for each individual or target vector $\mathbf{x}_i(t)$, a mutant vector $\mathbf{z}_i(t)$ is generated through mutation. The five mutation strategies that are referred to the most frequently are as follows:

$$\text{DE/rand/1}: \mathbf{z}_i(t) = \mathbf{x}_{R_1}(t) + F \cdot \left(\mathbf{x}_{R_2}(t) - \mathbf{x}_{R_3}(t)\right), \quad (3)$$

$$\text{DE/best/1}: \mathbf{z}_i(t) = \mathbf{x}_{best}(t) + F \cdot (\mathbf{x}_{R_1}(t) - \mathbf{x}_{R_2}(t)), \quad (4)$$

$$\begin{aligned}\text{DE/current-to-best/1}: \mathbf{z}_i(t) &= \mathbf{x}_i(t) + F \cdot (\mathbf{x}_{best}(t) - \mathbf{x}_i(t)) \\ &\quad + F \cdot (\mathbf{x}_{R_1}(t) - \mathbf{x}_{R_2}(t)),\end{aligned} \quad (5)$$

$$\begin{aligned}\text{DE/best/2}: \mathbf{z}_i(t) &= \mathbf{x}_{best}(t) + F \cdot (\mathbf{x}_{R_1}(t) - \mathbf{x}_{R_2}(t)) \\ &\quad + F \cdot \left(\mathbf{x}_{R_3}(t) - \mathbf{x}_{R_4}(t)\right),\end{aligned} \quad (6)$$

and

$$\begin{aligned}\text{DE/rand/2}: \mathbf{z}_i(t) &= \mathbf{x}_{R_1}(t) + F \cdot \left(\mathbf{x}_{R_2}(t) - \mathbf{x}_{R_3}(t)\right) \\ &\quad + F \cdot \left(\mathbf{x}_{R_4}(t) - \mathbf{x}_{R_5}(t)\right).\end{aligned} \quad (7)$$

The indices R_1, R_2, R_3, R_4, and R_5 are mutual integers randomly generated within the range [1, N] and are randomly generated once for each mutant vector. The scale factor F is a positive control parameter for scaling the difference vector, and $\mathbf{x}_{best}(t)$ is the vector with the best fitness value in the population at iteration t.

1.2. Crossover

After mutation, a binomial crossover operation forms the trial vector $\mathbf{u}(t) = [u^1(t), u^2(t), \ldots, u^D(t)]$:

$$u^d(t) = \begin{cases} z_i^d(t) & \text{if } d = K \text{ or } rand(0,1) \le CR_i \\ x_i^d(t) & otherwise \end{cases}, \quad (8)$$

where K is an integer randomly chosen from 1 to D, and parameter CR_i is called the crossover probability.

1.3. Selection

The selection operation selects the better of the target vector $\mathbf{x}_i$ and trial vector $\mathbf{u}_i$ according to their fitness values $f(\bullet)$. In a minimization problem, the selected vector is given by

$$\mathbf{x}_i(t+1) = \begin{cases} \mathbf{u}(t) & \text{if } f(\mathbf{u}(t)) \le f(\mathbf{x}_i(t)) \\ \mathbf{x}_i(t) & otherwise \end{cases}. \quad (9)$$

Table 1
Historical memory of M_{CR} and M_F.

Index	1	2	...	H-1	H
M_{CR}	$M_{CR,1}$	$M_{CR,2}$	...	$M_{CR,H-1}$	$M_{CR,H}$
M_F	$M_{F,1}$	$M_{F,2}$	...	$M_{F,H-1}$	$M_{F,H}$

2. Background

2.1. Success-history-based adaptive DE (SHADE)

Zhang and Sanderson (2009) achieved excellent improvements in the DE algorithm, and they presented a DE variant called adaptive differential evolution algorithm with optional external archive (JADE). SHADE, which is an improved version of JADE,

Tanabe & Fukunaga (2013) follows the general procedure of an evolutionary algorithm and uses a different parameter adaptation mechanism based on a historical record of successful parameter settings. Out of the 21 algorithms that were included in the CEC2013 competition on real parameter single-objective optimization (Liang, Qu, Suganthan, & Hernández-Díaz, 2013), SHADE ranked third, with the first two ranks being held by non-DE-based algorithms (Das, Mullick, & Suganthan, 2016).

2.1.1. Mutation

The mutation used by JADE and SHADE, current-to-pbest/1, is a variant of the current-to-best/1 strategy, where the greediness can be adjusted by using a parameter p:

$$\mathbf{z}_i(t+1) = \mathbf{x}_i(t) + F_i \cdot (\mathbf{x}_b(t) - \mathbf{x}_i(t)) + F_i \cdot (\mathbf{x}_{R_1}(t) - \tilde{\mathbf{x}}_{R_2}(t)). \quad (10)$$

The individual $\mathbf{x}_b$(t) is randomly chosen as one of the top 100p% individuals in the current population (denoted as P) with $p \in (0, 1]$. At the tth iteration, the individuals in the population are sorted according to their fitness values. The top 100p% individuals are those with the best values, and the number of these individuals is $N \cdot p$, where N is the total number of individuals in the population. The greediness of current-to-pbest/1 depends on p to balance exploitation and exploration (small p yields greedier behavior). In SHADE, each individual $\mathbf{x}_i$ has an associated p_i, which is set according to the following equation by generation:

$$p_i = rand(p_{\min}, 0.2), \quad (11)$$

where $p_{\min} = 2/N$. The maximum value 0.2 is the maximum value of the range of p.

To maintain diversity, JADE and SHADE use an optional, external archive (denoted as A). $\tilde{x}_{R_2}$ in (10) is selected from P∪A, the union of the population P and the archive A.

F_i is the mutation factor associated with $\mathbf{x}_i$ and is re-generated at each generation by the adaptation process introduced later.

2.1.2. Parameter adaptation

Each individual $\mathbf{x}_i$ is associated with its own CR_i and F_i parameters and generates trial vectors according to these values. S_{CR} and S_F denote the sets of all successful crossover probabilities of CR_i and F_i, respectively, at iteration t. In SHADE, the mean values of S_{CR} and S_F for each generation are stored in a historical memory (as M_{CR} and M_F, respectively) with H entries, as shown in Table 1.

In each generation, the control parameters CR_i and F_i used by each individual $\mathbf{x}_i$ are generated by first selecting an index R_i randomly from [1, H] and then applying the following equations:

$$CR_i = randn_i(M_{CR,R_i}, 0.1) \quad (12)$$

and

$$F_i = randc_i(M_{F,R_i}, 0.1), \quad (13)$$

where $randn_i(\mu, \sigma^2)$ and $randc_i(\mu, \sigma^2)$ are values selected randomly from normal and Cauchy distributions, respectively, with mean μ and variance σ^2.

The CR_i and F_i values used by successful individuals are recorded in S_{CR} and S_F, respectively, and at the end of the generation, the memory contents are respectively updated as follows:

$$M_{CR,k,t+1} = \begin{cases} \text{mean}_{WA}(S_{CR}) & \text{if } S_{CR} \neq 0 \\ M_{CR,k,t} & otherwise \end{cases} \tag{14}$$

and

$$M_{F,k,t+1} = \begin{cases} \text{mean}_{WL}(S_F) & \text{if } S_F \neq 0 \\ M_{F,k,t} & otherwise \end{cases}. \tag{15}$$

An index k, which is initialized to 1, determines the memory position to update. k is incremented whenever a new element is inserted into the history and it is set to 1 when $k > H$.

The weighted mean $\text{mean}_{WA}(S_{CR})$ and weighted Lehmer mean $\text{mean}_{WL}(S_F)$ are computed as follows:

$$\text{mean}_{WA}(S_{CR}) = \sum_{j=1}^{|S_{CR}|} w_j \cdot S_{CR,j} \tag{16}$$

and

$$\text{mean}_{WL}(S_F) = \frac{\sum_{j=1}^{|S_F|} w_j \cdot S_{F,j}^2}{\sum_{j=1}^{|S_F|} w_j \cdot S_{F,j}}, \tag{17}$$

where

$$w_j = \frac{\Delta f_j}{\sum_{j=1}^{|S_{CR}|} \Delta f_j} \tag{18}$$

and

$$\Delta f_j = \left| f\left(\mathbf{u}_j(t)\right) - f(\mathbf{u}_k(t)) \right|. \tag{19}$$

2.2. Multi-cross operation (MCO)

Inspired by the swarming behavior of flocking birds, Kennedy and Eberhart (1995) introduced the PSO algorithm. Each particle in PSO represents a possible solution for an optimization problem in a D-dimensional space and has a velocity vector $\mathbf{v}_i$ and position vector $\mathbf{xp}_i$. Each particle flies within the search space and is attracted by its own previous best position ($pbest_i$) and by the global best position of the swarm ($gbest$). The velocity and position for the ith ($i = 1, 2, \ldots, N$) particle are respectively updated as follows during the tth ($t = 1, 2, \ldots, T$) iteration:

$$v_i^d(t+1) = wv_i^d(t) + c_1 r_1 \left(p_i^d(t) - y_i^d(t)\right) + c_2 r_2 (g^d(t) - xp_i^d(t)) \tag{20}$$

and

$$xp_i^d(t+1) = xp_i^d(t) + v_i^d(t+1), \tag{21}$$

where p_i is $pbest_i$; g is $gbest$; N is the population size; d ($d = 1, 2, \ldots, D$) denotes the dimension of the solution space; T is a predefined maximum number of iterations; c_1 and c_2 are the acceleration coefficients indicating the influence of the $pbest_i$ and $gbest$ values of the particle, respectively; and r_1 and r_2 are two uniformly distributed random numbers within the range of [0, 1]. The parameter w is the inertial weight (Kennedy, 1999).

Chang (2007) proposed a multi-crossover genetic approach to solve the controller design problem. A multi-crossover uses three parent chromosomes (i.e., Θ_1, Θ_2, and Θ_3), unlike a classical crossover, which uses only two chromosomes. Chromosome Θ_1 is selected as the premier parent if it has the best fitness value. The following equation is then be used to generate a new chromosome:

$$\Theta_1' = \Theta_1 + r(2\Theta_1 - \Theta_2 - \Theta_3), \tag{22}$$

where r is a uniformly distributed random number within the range of [0, 1].

High-exploration PSO (HEPSO) (Mahmoodabadi, Mottaghi, & Bagheri, 2014) uses this idea and involves a new PSO operation that employs $gbest$ as the premier parent and $pbest_i$ as the second parent:

$$v_i^d(t+1) = r(0.5 \times c \times g^d(t) - p_i^d(t) - xp_i^d(t)), \tag{23}$$

where c is a social learning factor similar to c_1 and c_2 in a PSO velocity update.

Hybrid non-parametric PSO (HNPPSO) (Liu, Ji, & Liu, 2018) also omits c:

$$v_i^d(t+1) = r(g^d(t) - p_i^d(t) - xp_i^d(t)). \tag{24}$$

MCO is characterized as having a high convergence speed and good local search ability. However, these features also lead to falling into a local optimum.

2.3. Hierarchical structure in SI variants

In its hierarchical structure, an algorithm is always divided into layers, and different tasks are performed by these layers. Dhahri, Alimi, and Abraham (2012) proposed a hierarchical DE algorithm composed of two layers for beta basis function neural network (BBFNN) design. The top layer searches for optimal networks in architecture space, and the DE algorithm in the bottom layer is used to train the BBFNN by adjusting the neural parameters. Dual-population DE (Zhong et al., 2013) involves two populations (two layers) with different purposes and a bidirectional migration operator. The first population is called the global population (GP). It focuses on global searching, with parameter settings and operators that effectively maintain the population diversity. The second population is called the local population (LP). It focuses on local searching, with parameter settings and operators that are suitable for local fine-tuning. The bidirectional migration operator exchanges the best individuals between the two populations accordingly, to ensure the high population diversity of the GP and the fast convergence of the LP simultaneously. However, the migration of best individuals from the LP to the GP may lead to the stagnation of the GP and may decrease the performance of the global search.

According to the search algorithms used in different levels, these hierarchical structure variants can also be classified as homogeneous schemes (Maio, Baronchelli, & Zio, 2014; Sreeja & Sankar, 2016) or heterogeneous schemes (Epitropakis, Plagianakos, & Vrahatis, 2012). In the former, the search algorithms in all of the levels are the same, and in the latter, different search algorithms are adopted in different levels. For instance, Crisscross-search PSO (CSPSO) (Meng, Li, Yin, Chen, & Guo, 2016) employs a two-layer evolution framework in which the bottom layer performs PSO and the top layer implements crisscross search optimization (CSO). However, the parameter p_v introduced by CSPSO is given as an empirical value. A guideline for choosing its value may be useful for real optimization problems in expert systems.

According to the number of layers in the scheme, the hierarchical structure can be designated as a two-layer (level) structure (Dhahri et al., 2012; Lim & Isa, 2014) or a multi-layer structure (Rastegar, Araújo, & Mendes, 2017; Wang, Yang, & Chen, 2014). Rastegar et al. (2017) introduced a hierarchical PSO structure that includes six levels that extract all of the fuzzy logic system parameters of a fuzzy model automatically. The particles in different levels represent variables, membership functions, parameters sets, etc. in fuzzy models. A multi-layer PSO (MLPSO) structure was also proposed (Wang et al., 2014). In MLPSO, a subswarm (subpopulation) in the lower level may be considered a particle in the immediate upper layer, and this subswarm is called a swarm particle. That is to say, a swarm particle in the upper layer contains some particles in the lower layer. In the lower layer, the swarm particles search a solution space on a small scale. In the upper layer, they search on a large scale. The swarm particle in the lower layer receives information not only from the current layer, but also from the upper layers. An obstacle for the multi-layer structure may be that implementing the algorithm is difficult because the structure is too complex, especially for users who are not experts in computer science.

Two-layer hierarchical structures can be further categorized into three groups according to the number of subswarms (subpopulations) in the top and bottom layers.

(i) One-to-one. In this scheme, there is one subswarm in the top layer and one subswarm in the bottom layer. The two subswarms in this strategy are called current and memory swarms in an adaptive two-layer PSO algorithm with an elitist learning strategy (Lim & Isa, 2014). The current swarm is the particle swarm that includes all of the particles, and the memory swarm includes all of the *pbest* positions. A two-way interaction between global best individuals in the two swarms is introduced to enhance the convergence capability. However, this scheme may lead to premature convergence. The two-layer structure was also adopted in ACO (Rusin & Zaitseva, 2012). Ants working in a fairly traditional way in the bottom layer are called workers, and those in the top layer are called ant managers. Each worker has its own plan and searches the unique environment, while the ant managers monitor groups of workers. Because of its concise structure, this one-to-one structure has been implemented widely by many researchers. Generally, the task of the bottom layer is a global search, and the top layer implements a local search. An information exchange scheme is necessary between two layers, especially for the best individuals.

(ii) One-to-multi (María, Paola, Germán, & Miguel, 2017). In this scheme, one subswarm is in the top layer and several subswarms are in the bottom layer. The subswarm in top layer processes information transferred from the bottom layer or manages the bottom layer. Peng and Lu (2013) proposed a two-layer hierarchical PSO model. For each subswarm in the bottom layer, particles move towards the optimum based on the comprehensive learning method (Liang, Qin, Suganthan, & Baskar, 2006). Each subswarm in the bottom layer generates a best particle that could enter the top layer. Thus, the number of particles in the top layer is identical to the number of subswarms in the bottom layer. However, the search behaviors of swarms in the bottom layer are implemented independently in this reference. In the future, imposing mutual communication among these swarms can be investigated in depth. In hierarchical PSO with ortho-cyclic circles (Ganapathy, Vaidehi, Kannan, & Murugan, 2014), the particles are partitioned into multiple subswarms called circles. The best particle in each circle is considered to depict the behavior of the respective circle and is called a circle representative (CR). The CRs are treated as high-layer particles, and the particles within the circles are treated as low-layer particles. However, the performance of this algorithm for rotated and shifted fitness functions is unclear. In the strategy proposed by Gumaida and Luo (2017), there are L subswarms in the bottom layer, where each swarm contains M particles. In the top layer, there is only one swarm with L particles, equal to the number of subswarms in the bottom layer. Each swarm in the bottom layer is taken as a particle in the top layer. Each global best location *gbest* of a subswarm in the bottom layer is taken as the local best location *pbest* in the top layer. Generally speaking, there are several subswarms in the bottom layer in the one-to-multi structure. How to share information among these subswarms is an important issue for this structure.

(iii) Multi-to-multi. In this scheme, the top and bottom layers both include several subswarms. In the scheme proposed by Ma, Chen, Hu, and Zhu (2014), each subpopulation employs the canonical ABC method to search for the optimum in parallel in the bottom level. That is, in each iteration, K subpopulations in the bottom level generate K best solutions, which are constructed into a complete solution species that updates the top level. In the top level, all of the individuals are divided into several species, and these species make up a multispecies community. An information exchange mechanism in the multispecies community was proposed based on a crossover operator, by which each species can learn from its neighborhood. Hu, Liu, and Zhang (2016) proposed a new co-evolutionary ABC algorithm based on a hierarchical communication model (HCM). The HCM involves communication on two levels: the species level (bottom level) and the group level (top level). One group in the top level is composed of one or several species from the bottom level. The species communicate with each other synchronously to increase the convergence speed. Meanwhile, the groups communicate with each other asynchronously to finish the global research. In the multi-to-multi structure, there are several subswarms in both the top and bottom layers. Therefore, this structure will lead to a high computational complexity, and it will not be easy for users to implement this structure for real problems in expert systems.

3. Hierarchical differential evolution combined with MCO (HDEMCO)

In this section, a modified MCO (MMCO) is proposed, and the individual position updates in MMCO are discussed. First-order stability analysis is also presented. Finally, a hierarchical DE algorithm combined with MMCO is proposed.

3.1. MMCO

Based on (20) and (22), the following MMCO operator with a velocity vector is proposed:

$$v_i^d(t+1) = wv_i^d(t) + c^d(g^d(t) - p_i^d(t) - xp_i^d(t)), \tag{25}$$

where c is the social learning factor similar to c_1 and c_2 in a PSO velocity update.

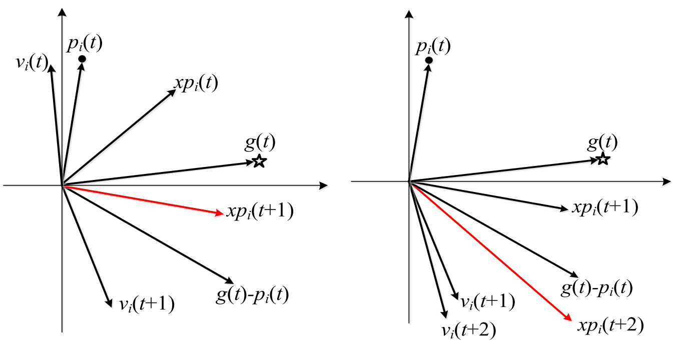
Unlike the MCO operators in HEPSO and HNPPSO, the MMCO operator proposed in this paper introduces a velocity term, just as in PSO. Since the initial value of the velocity term is generally random, the introduction of this term makes MMCO less likely to fall into local optima on the premise of guaranteeing fast convergence.

3.1.1. Individual position updates in MMCO

To provide a more intuitive description of the individual position updates in MMCO, an example is given in Fig. 1. It can be seen that, in each iteration, the position of the individual is kept close to $g(t)$-$p_i(t)$, thus avoiding the problem of “oscillation” between *gbest* and $pbest_i$ that occurs in PSO (Liang et al., 2006). In fact, according to the stability analysis in Section 3.1.2, the positions of the individuals will converge to $g(t)-p_i(t)$ when certain conditions are met.

3.1.2. First-order stability analysis of MMCO without stagnation assumption

In PSO stability analysis, it is usually assumed that $pbest_i$ and *gbest* are not updated during the run (stagnation assumption) to



(a) tth iteration (b) (t+1)th iteration

Fig. 1. Individual position updates in MMCO with $w = 0.6$ and $c = 1.0$.

simplify the formulation of update rules (Bonyadi & Michalewicz, 2016; Liu et al., 2018). To find the convergence regions under more general conditions, the MMCO stability analysis in this study was performed without the stagnation assumption, according to the method proposed by Bonyadi and Michalewicz (2016). In the calculations, $pbest_i$ and $gbest$ were considered to be random variables updated during the run with arbitrary means and variances.

According to (25), the position updates in MMCO are conducted in each dimension separately. Hence, analysis in 1-D space is generalizable to multidimensional space (Bonyadi & Michalewicz, 2017). Given this characteristic, the vector notation is dropped from the following equations. Thus, (21) and (25) can be respectively rewritten as

$$xp_{t+1} = xp_t + v_{t+1} \tag{26}$$

and

$$v_{t+1} = wv_t + c(g_t - p_t - xp_t). \tag{27}$$

Considering (26), (27) can be rewritten as

$$xp_{t+1} - xp_t = w(xp_t - xp_{t-1}) + c(g_t - p_t - xp_t), \tag{28}$$

then

$$xp_{t+1} = (1 + w - c)xp_t - w \cdot xp_{t-1} + c(g_t - p_t). \tag{29}$$

First-order stability for a 1-D stochastic sequence $\{y_1, y_2, \ldots\}$ (where y_t is a random variable for all values of t) is defined as follows:

Definition 1. A sequence $\{y_1, y_2, \ldots\}$ is called "first-order stable" if and only if the sequence $\{E(y_1), E(y_2), \ldots\}$, where $E(.)$ is the expectation operation, converges.

The expected position of a particle can be calculated by applying the expectation operator to both sides of (29), i.e.,

$$E(xp_{t+1}) = E(l)E(xp_t) - \mu_w E(xp_{t-1}) + \mu_c\mu_g - \mu_c\mu_p, \tag{30}$$

where μ_w, μ_c, μ_g, and μ_p are the expected values of w, c, g, and p, respectively; $l = 1 + w - c$; and $E(l) = 1 + \mu_w - \mu_c$.

Then, (30) can be rewritten in matrix form (Bonyadi & Michalewicz, 2016) as

$$\vec{y}_{t+1} = M\vec{y}_t + \vec{b}, \tag{31}$$

where

$$\vec{y}_t = \begin{bmatrix} E(xp_t) & E(xp_{t-1}) \end{bmatrix}^T$$
$$M = \begin{bmatrix} E(l) & -\mu_w \\ 1 & 0 \end{bmatrix}$$
$$\vec{b} = \begin{bmatrix} \mu_c\mu_g - \mu_c\mu_p & 0 \end{bmatrix}^T. \tag{32}$$

The eigenvalues of M are

$$\lambda_{1,2} = \frac{-E(l) \pm \sqrt{E^2(l) - 4\mu_w}}{2}. \tag{33}$$

Therefore, the expectation of xp_t is convergent if and only if (Bonyadi & Michalewicz, 2016)

$$-1 < \mu_w < 1 \text{ and } 0 < \mu_c < 2(\mu_w + 1). \tag{34}$$

Let us assume that the expected particle position is convergent and that it converges to a value E_{xp}. Then,

$$E_{xp} = E(l)E_{xp} - \mu_w E_{xp} + \mu_c\mu_g - \mu_c\mu_p, \tag{35}$$

and E_{xp} can be calculated using

$$E_{xp} = \frac{\mu_c\mu_g - \mu_c\mu_p}{1 - E(l) + \mu_w} = \mu_g - \mu_p. \tag{36}$$

Therefore, when (34) is satisfied, the particles are first-order stable, and their expected positions converge to u_g-u_p after certain iterations.

3.2. HDEMCO

In the proposed method, the updates of individuals by MMCO are implemented based on the SHADE results. That is to say, $pbest_i$ in MMCO shown in (25) is replaced by $\mathbf{x}_i$ in SHADE:

$$v_i^d(t+1) = wv_i^d(t) + c^d(g^d(t) - x_i^d(t) - xp_i^d(t)). \tag{37}$$

Algorithm 1 presents the entirety of the proposed evolution process.

Algorithm 1 Pseudocode of the proposed HDEMCO algorithm.

```
Initialize x_i, xp_i, v_i, fitness values, and gbest of the population;
for t = 1:T
  for i = 1:N
    update the position of the ith individual (x_i) in the bottom layer by
    SHADE with p = 0.5;
    if(fitness(x_i) < fitness(gbest))
      gbest = x_i
    end if
    update the position of the ith individual (xp_i) in the top layer by using
    (21) and (37);
    if(fitness(xp_i) < fitness(gbest))
      gbest = xp_i
    end if
  end for
end for
```

The proposed structure has two layers: the top layer is MCO, and the bottom layer is SHADE. As described in HNPPSO, MCO provides fast convergence. Meanwhile, SHADE has better global search performance. Thus, the proposed algorithm combines the advantages of the different algorithms.

In the framework proposed in this paper, the MCO search in the top layer is based on SHADE in the bottom layer, and the MCO results only affect $gbest$. According to the SHADE strategy, $gbest$ does not affect the SHADE search. That is to say, the MCO search results do not affect the bottom layer. Because MCO provides faster convergence, it can easily fall into local optima. If the MCO search results affected the bottom layer, the whole algorithm would complete prematurely. Therefore, this algorithm can balance the global and local searches.

In JADE/SHADE, the greediness of current-to-pbest/1 depends on the parameter p to balance exploitation and exploration. To make the SHADE algorithm focused on the global search, p was set to 0.5 in the SHADE algorithm in this study to provide the $\mathbf{x}_b(t)$ selection with a larger scope and higher randomness.

Compared to other structures, the proposed algorithm has the following advantages:

Table 2
Parameter settings of the compared algorithms.

Algorithm	Parameter settings	Reference
JADE	$p = 0.05$, $c = 0.1$	Zhang & Sanderson, 2009
SHADE	$c = 0.1$	Tanabe & Fukunaga, 2013
MPEDE	$\lambda_1 = \lambda_2 = \lambda_3 = 0.2$, $c = 0.1$	Wu et al., 2016
SLADE	–	Zhao et al., 2016
pbestrrJADE	$p = 0.05$, $c = 0.1$	Yi et al., 2016
HEPSO	$c = 2.0$, $C_{1i} = 2.5$, $C_{1f} = 0.5$, $C_{2i} = 0.5$, $C_{2f} = 2.5$	Mahmoodabadi et al., 2014
CSPSO	w: $0.9 - 0.4$, $c_1 = c_2 = c = 2.0$, $P_v = 0.6$	Meng et al., 2016
HNPPSO	–	Liu et al., 2018
HDEMCO	w: $0.9 - 0.4$, $c = 1.0$	–

(1) It combines the advantages of the different algorithms, i.e., MCO and SHADE.
(2) The search results in the top layer do not affect the bottom layer. Therefore, the bottom layer can avoid premature results and focuses on the global search, while the top layer focuses on the local search. Then, the proposed algorithm can balance the global and local searches.
(3) The MMCO stability was analyzed in this study without the stagnation assumption. Therefore, in contrast with HNPPSO, the proposed algorithm can find the convergence regions under more general conditions.

The proposed algorithm also presents some weaknesses:

(1) In contrast with classical PSO and DE, there are two layers with different types of algorithm. Therefore, the implementation of the proposed structure is not as easy for users who are not experts in computer science.
(2) Since the proposed algorithm implements the bottom layer using SHADE, the parameters in SHADE are also reserved in the proposed algorithm. On the other hand, a parameter w is introduced in the MMCO for top layer. How to tune these parameters for real optimization problems in expert systems should be studied carefully in the future.
(3) Only the first-order stability is analyzed in the proposed algorithm. However, a second-order stability analysis is also necessary, which represents another weakness.

4. Experiments and discussion

In this section, the computational results of HDEMCO are discussed along with comparisons with other eight state-of-the-art algorithms and five other SI algorithms.

4.1. Test suites

The performance of the proposed HDEMCO algorithm was tested on 30 benchmark functions proposed in the CEC 2017 (Awad, Ali, Liang, Qu, & Suganthan, 2017) special session on single-objective real-parameter optimization. These 30 test functions can be divided into the following classes:

(1) Unimodal functions F1–F3;
(2) Simple multimodal functions F4–F10;
(3) Hybrid functions F11–F20;
(4) Composition functions F21–F30.

4.2. Experimental settings

We compared the proposed HDEMCO algorithm with eight advanced DE and PSO variants. Table 2 describes the parameter settings for all of the variants, which were extracted from their corresponding studies.

In this study, a modified SHADE algorithm was implemented in the bottom layer, and SHADE is an improved version of JADE. JADE and SHADE are also two representative DE variants that are very efficient and are frequently cited in literature. Therefore, these two algorithms were compared in our experiments. In addition, there are three subswarms in MPEDE Wu, Mallipeddi, Suganthan, Wang, & Chen, 2016), and one subswarm is implemented in JADE. Therefore, MPEDE can also be viewed as an improved version of JADE. SLADE (Zhao, Yang, Hu, & Che, 2016) is a recently published improved version of DE, which also adopts an adaptive parameter control strategy similar to SHADE. pbestrrJADE (Yi, Zhou, Gao, Li, & Mou, 2016) is a recently published improved version of JADE. In our proposed structure, the MMCO algorithm implemented in the top layer includes a velocity vector, similarly to PSO. Therefore, we also compared three PSO variants in our experiments: HEPSO (Mahmoodabadi et al., 2014) employs the MCO shown in ((23); CSPSO (Meng et al., 2016) is a recently published PSO variant and adopts a two-layer structure similar to that in our proposed method; and HNPPSO (Liu et al., 2018) is another recently published method and also adopts MCO.

Table 3
Parameter settings of the SI algorithms.

Algorithm	Parameter settings	Reference
GA	crossover probability = 0.35 mutation probability = 0.15	Nazarahari et al., 2019
DE	$F = 0.5$, $\mathrm{Cr} = 0.9$	Cai et al., 2018
PSO	w: $0.9 - 0.4$, $c1 = c2 = 1.49445$	Shi & Eberhart, 1998
ACO	$\rho = 0.1$	Chen et al., 2017
CS	$\mathrm{Pa} = 0.25$	Yang & Deb, 2010

Besides these eight variants, we also compared HDEMCO with five other SI algorithms, namely, GA, DE, PSO, ACO, and cuckoo search (CS). CS is a new population-based metaheuristic search algorithm developed by Yang and Deb (2010). Table 3 describes the parameter settings for all of the SI algorithms.

For fair comparison, we set the dimension of the solution space to $D = 10$, 30, and 50 for all of the test functions, and we set the population size N to 100. We set the maximum fitness evaluation MaxFEs to 3×10^5 for all of the problems and independently repeated each test 51 times to reduce the statistical errors. In the experiments, the algorithms terminated upon reaching MaxFEs or when the error was smaller than 10^{-8}, as recommended by CEC2017 (Awad et al., 2017). The error is the difference between the optimum solution and the solution obtained by the algorithm variants.

4.3. Solution accuracy comparison

The experimental results for the $D = 10$, $D = 30$, and $D = 50$ problems are shown in Tables 4–6, respectively. These tables provide the mean values and standard deviations obtained using the HDEMCO and other variants considered herein for all 30 benchmark functions. The mean values and standard deviations averaged over 51 independent iterations are listed in each row of the tables.

Table 4
Means and standard deviations of the error for the $D = 10$ problems. The best entries appear in bold.

Functions		JADE	SHADE	MPEDE	SLADE	pbestrrJADE	HEPSO	CSPSO	HNPPSO	HDEMCO
F1	Mean	**0.00E+000**	**0.00E+000**	**0.00E+000**	**0.00E+000**	**0.00E+000**	7.65E−001	7.00E+001	**0.00E+000**	**0.00E+000**
	Std.	**0.00E+000**	**0.00E+000**	**0.00E+000**	**0.00E+000**	**0.00E+000**	1.75E+000	1.10E+002	**0.00E+000**	**0.00E+000**
	+/=/−	=	=	=	=	=	+	+	=	
F2	Mean	**0.00E+000**	**0.00E+000**	**0.00E+000**	**0.00E+000**	**0.00E+000**	3.54E−008	5.31E−007	2.11E−006	**0.00E+000**
	Std.	**0.00E+000**	**0.00E+000**	**0.00E+000**	**0.00E+000**	**0.00E+000**	3.05E−008	4.60E−007	2.71E−006	**0.00E+000**
	+/=/−	=	=	=	=	=	+	+	+	
F3	Mean	**0.00E+000**	**0.00E+000**	**0.00E+000**	**0.00E+000**	**0.00E+000**	**0.00E+000**	**0.00E+000**	**0.00E+000**	**0.00E+000**
	Std.	**0.00E+000**	**0.00E+000**	**0.00E+000**	**0.00E+000**	**0.00E+000**	**0.00E+000**	**0.00E+000**	**0.00E+000**	**0.00E+000**
	+/=/−	=	=	=	=	=	=	=	=	
F4	Mean	**0.00E+000**	**0.00E+000**	**0.00E+000**	**0.00E+000**	3.52E−003	4.72E−007	3.07E−007	**0.00E+000**	**0.00E+000**
	Std.	**0.00E+000**	**0.00E+000**	**0.00E+000**	**0.00E+000**	1.70E−002	3.63E−007	5.05E−007	**0.00E+000**	**0.00E+000**
	+/=/−	=	=	=	=	+	+	+	=	
F5	Mean	**0.00E+000**	6.63E−001	4.25E+000	1.97E+000	**0.00E+000**	**0.00E+000**	2.55E+000	1.53E+000	1.56E−001
	Std.	**0.00E+000**	7.62E−001	1.67E+000	1.57E+000	**0.00E+000**	**0.00E+000**	9.40E−001	9.95E−001	3.65E−001
	+/=/−	−	+	+	+	−	−	+	+	
F6	Mean	**0.00E+000**	**0.00E+000**	**0.00E+000**	3.19E−007	**0.00E+000**	**0.00E+000**	**0.00E+000**	**0.00E+000**	**0.00E+000**
	Std.	**0.00E+000**	**0.00E+000**	**0.00E+000**	1.60E−006	**0.00E+000**	**0.00E+000**	**0.00E+000**	**0.00E+000**	**0.00E+000**
	+/=/−	=	=	=	+	=	=	=	=	
F7	Mean	1.01E+001	1.11E+001	1.35E+001	1.08E+001	**1.00E+001**	8.76E+000	1.32E+001	1.50E+002	1.07E+001
	Std.	1.22E+000	4.84E−001	1.50E+000	1.99E+000	**1.72E+000**	3.81E+000	8.86E−001	7.21E+000	3.81E−001
	+/=/−	−	+	+	=	−	−	+	+	
F8	Mean	**0.00E+000**	6.43E−001	4.29E+000	2.78E+000	**0.00E+000**	**0.00E+000**	2.46E+000	1.98E+000	2.92E−001
	Std.	**0.00E+000**	6.84E−001	1.57E+000	3.59E+000	**0.00E+000**	**0.00E+000**	8.03E−001	1.20E+000	5.73E−001
	+/=/−	−	+	+	+	−	−	+	+	
F9	Mean	**0.00E+000**	**0.00E+000**	**0.00E+000**	4.81E−002	**0.00E+000**	**0.00E+000**	**0.00E+000**	**0.00E+000**	**0.00E+000**
	Std.	**0.00E+000**	**0.00E+000**	**0.00E+000**	1.68E−001	**0.00E+000**	**0.00E+000**	**0.00E+000**	**0.00E+000**	**0.00E+000**
	+/=/−	=	=	=	+	=	=	=	=	
F10	Mean	**1.06E−001**	1.04E+000	1.59E+002	4.27E+001	4.19E−001	1.37E+001	1.03E+001	9.75E+000	1.21E+000
	Std.	**6.30E−002**	1.76E+000	1.28E+002	7.33E+001	6.75E−001	2.03E+001	5.91E+000	1.77E+001	2.82E+000
	+/=/−	−	−	+	+	−	+	+	=	
F11	Mean	**0.00E+000**	**0.00E+000**	1.56E−001	2.11E+000	5.15E−008	**0.00E+000**	3.20E−001	1.75E−001	**0.00E+000**
	Std.	**0.00E+000**	**0.00E+000**	3.65E−001	2.36E+000	8.30E−008	**0.00E+000**	6.65E−001	3.83E−001	**0.00E+000**
	+/=/−	=	=	+	+	+	=	+	+	
F12	Mean	5.57E+001	7.23E+001	4.54E+001	4.21E+002	**2.57E+001**	2.61E+003	2.30E+003	5.58E+003	3.11E+001
	Std.	7.53E+001	7.22E+001	6.53E+001	1.97E+002	**5.56E+001**	2.13E+003	1.26E+003	2.66E+003	5.26E+001
	+/=/−	+	+	+	+	−	+	+	+	
F13	Mean	6.41E−001	2.83E+000	3.89E+000	7.37E+000	**4.47E−001**	1.03E+000	4.74E+001	4.79E+003	7.02E−001
	Std.	1.64E+000	2.46E+000	2.06E+000	5.89E+000	**1.43E+000**	2.40E+000	5.64E+001	2.10E+003	1.67E+000
	+/=/−	−	+	+	+	−	+	+	+	
F14	Mean	**0.00E+000**	**0.00E+000**	5.26E−001	7.26E+000	**0.00E+000**	**0.00E+000**	3.51E−001	6.94E−001	**0.00E+000**
	Std.	**0.00E+000**	**0.00E+000**	8.52E−001	9.87E+000	**0.00E+000**	**0.00E+000**	4.58E−001	6.37E−001	**0.00E+000**
	+/=/−	=	=	+	+	=	=	+	+	
F15	Mean	3.10E−004	1.20E−002	1.32E−001	2.18E+000	**2.86E−004**	9.17E−003	4.76E−001	9.05E−001	2.45E−003
	Std.	3.44E−004	4.57E−002	2.20E−001	5.96E+000	**2.70E−004**	1.36E−002	3.95E−001	6.72E−001	1.07E−002
	+/=/−	−	+	+	+	−	+	+	+	
F16	Mean	4.09E−004	3.85E−002	1.88E+000	2.20E+001	**2.22E−004**	6.77E−002	1.85E−001	6.14E−001	2.71E−002
	Std.	5.03E−004	1.08E−001	3.77E+000	3.76e+001	**2.46E−004**	1.84E−001	1.30E−001	1.83E+000	9.38E−002
	+/=/−	−	=	+	+	−	+	+	+	
F17	Mean	**0.00E+000**	7.73E−004	2.71E−001	1.19E+001	**0.00E+000**	8.33E−003	4.45E−001	6.40E+000	**0.00E+000**
	Std.	**0.00E+000**	3.86E−003	3.14E−001	1.23e+001	**0.00E+000**	1.00E−002	3.47E−001	7.78E+000	**0.00E+000**
	+/=/−	=	=	+	+	=	+	+	+	
F18	Mean	5.90E−003	1.24E−001	1.66E−001	1.52E+001	**2.52E−004**	1.08E−002	7.08E+000	3.83E+002	1.88E−001
	Std.	3.93E−002	1.78E−001	2.60E−001	1.09e+001	**3.90E−004**	9.14E−003	5.15E+000	5.47E+002	2.03E−001
	+/=/−	−	=	=	+	−	+	+	+	
F19	Mean	2.94E−002	**0.00E+000**	1.06E−002	8.84E−001	**0.00E+000**	6.14E−007	2.24E−002	1.51E+001	**0.00E+000**
	Std.	2.10E−001	**0.00E+000**	1.10E−002	1.03e+000	**0.00E+000**	3.88E−06	1.05E−002	7.07E+001	**0.00E+000**
	+/=/−	+	=	+	+	=	+	+	+	
F20	Mean	2.71E+002	**0.00E+000**	1.83E−002	7.31E+000	**0.00E+000**	**0.00E+000**	**0.00E+000**	2.13E−001	**0.00E+000**
	Std.	1.18E+002	**0.00E+000**	7.41E−002	9.36e+000	**0.00E+000**	**0.00E+000**	**0.00E+000**	4.26E−001	**0.00E+000**
	+/=/−	+	=	=	+	=	=	=	+	
F21	Mean	1.37E+002	1.32E+002	1.41E+002	1.62E+002	1.30E+002	**4.51E+001**	7.45E+001	1.00E+002	1.02E+002
	Std.	4.97E+001	4.80E+001	5.57E+001	5.21e+001	4.70E+001	**4.63E+001**	4.19E+001	1.07E+000	6.28E+000
	+/=/−	+	+	+	+	+	−	−	=	
F22	Mean	9.32E+001	9.43E+001	9.04E+001	9.75E+001	9.47E+001	**1.37E+001**	6.42E+001	9.83E+001	1.00E+002
	Std.	2.42E+001	2.28E+001	2.93E+001	1.48e+001	2.16E+001	**3.47E+001**	3.79E+001	1.24E+001	0.00E+000
	+/=/−	−	−	−	=	−	−	−	−	

(*continued on next page*)

Table 4 (*continued*)

Functions		JADE	SHADE	MPEDE	SLADE	pbestrrJADE	HEPSO	CSPSO	HNPPSO	HDEMCO
F23	Mean	3.00E+002	3.00E+002	3.06E+002	3.07E+002	2.94E+002	1.82E+002	**2.55E+000**	2.93E+002	3.00E+002
	Std.	4.23E−001	8.86E−001	2.61E+000	3.00e+000	4.20E+001	1.47E+002	**3.95E+000**	2.86E+001	3.60E−001
	+/=/−	=	=	=	+	−	−	−	−	
F24	Mean	2.49E+002	2.81E+002	2.83E+002	3.26E+002	2.48E+002	**9.01E+001**	9.22E+001	1.01E+002	2.15E+002
	Std.	1.06E+002	8.00E+001	9.72E+001	4.65e+001	1.09E+002	**3.00E+001**	1.76E+001	1.40E+001	1.02E+002
	+/=/−	+	+	+	+	+	−	−	−	
F25	Mean	4.14E+002	4.12E+002	4.14E+002	4.21E+002	4.14E+002	**1.72E+002**	4.03E+002	4.34E+002	4.18E+002
	Std.	2.22E+001	2.15E+001	2.21E+001	2.44e+001	2.20E+001	**1.27E+002**	1.35E+001	1.89E+001	2.28E+001
	+/=/−	=	=	=	+	=	−	=	+	
F26	Mean	2.98E+002	3.00E+002	3.00E+002	3.51E+002	3.00E+002	**2.35E+001**	2.49E+002	2.90E+002	2.88E+002
	Std.	1.40E+001	0.00E+000	0.00E+000	4.24e+001	4.25E−014	**6.50E+001**	7.13E+001	3.00E+001	5.88E+001
	+/=/−	+	+	+	+	+	−	−	=	
F27	Mean	3.87E+002	3.90E+002	3.89E+002	3.91E+002	3.87E+002	**1.59E+002**	3.87E+002	3.95E+002	3.89E+002
	Std.	8.42E−001	2.21E+000	2.14E−001	1.96e+000	7.53E−001	**1.22E+002**	1.29E+000	2.89E+000	1.64E+000
	+/=/−	=	=	=	=	=	−	−	=	
F28	Mean	3.76E+002	3.52E+002	3.47E+002	4.67E+002	3.39E+002	**7.05E+001**	2.87E+002	3.59E+002	3.09E+002
	Std.	1.45E+002	1.14E+002	1.12E+002	1.30e+002	1.00E+002	**1.28E+002**	4.48E+001	1.08E+002	4.43E+001
	+/=/−	+	+	+	+	+	−	−	+	
F29	Mean	2.31E+002	2.30E+002	2.31E+002	2.35E+002	2.33E+002	**2.19E+002**	2.38E+002	2.48E+002	2.29E+002
	Std.	1.64E+000	3.13E+000	3.41E+000	8.34e+000	1.70E+000	**2.92E+001**	3.68E+000	9.03E+000	2.40E+000
	+/=/−	+	=	+	+	+	=	+	+	
F30	Mean	4.48E+002	4.11E+002	3.24E+004	2.25E+005	5.12E+002	9.09E+002	1.27E+003	1.10E+003	**3.98E+002**
	Std.	2.63E+001	3.43E+001	1.60E+005	3.96e+005	4.80E+001	1.64E+002	4.50E+002	1.39E+003	**1.01E+001**
	+/=/−	+	+	+	+	+	+	+	+	
w/t/l	7/14/9	10/18/2	17/12/1	23/7/0	7/12/11	12/7/11	18/5/7	19/8/3		

A two-sided Wilcoxon rank sum test (Wilcoxon, 1945) at the 0.05 significance level was used to evaluate the statistical significance of the differences between HDEMCO and every other algorithm. The null hypothesis H_0 was that the compared results were independent from the identical continuous distributions. If the *p*-value was less than 0.05, then there was considered to be a significant difference between the two algorithms, and the alternative hypothesis H_1 was accepted. The "+" signs signify the cases in which HDEMCO exhibited superior performance, whereas the "−" signs indicate inferior performance. In such cases, H_0 was rejected. The cases marked with "=" signs are those for which the performance difference was not statistically significant.

The Friedman test (Derrac, García, Molina, & Herrera, 2011) is a non-parametric multiple comparison test, which was used in this study to test the differences between the nine compared algorithms (including HDEMCO). Table 7 shows the results. The last columns in the table denote the corresponding measured *p*-values, which suggest the significant differences between the compared algorithms at the 0.05 significance level. The Friedman test assigned the lowest rank to the best-performing algorithm.

4.3.1. Performances for the 10-D problems

The experimental results for the $D = 10$ problems are shown in Table 4. Since the dimension for these problems was low, some algorithms could find the global optimum for many functions. For instance, HDEMCO, JADE, and pbestrrJADE could find the global optimum for 11 functions.

According to the average ranking shown in Table 7, HDEMCO obtained the third place among all of the variants (rank = 3.83), while the best and the second-best rankings were achieved by HEPSO (rank = 3.52) and pbestrrJADE (rank = 3.58) respectively. In fact, JADE surpasses HDEMCO according to Wilcoxon rank sum test results, even though its average ranking (rank = 4.0) is worse than that of HDEMCO. According to the experimental results, HDEMCO performed better than JADE for seven functions and yielded similar performance for 14 functions and inferior performance for nine functions. HEPSO and HDEMCO obtained the global optimum for eight and 11 functions, respectively. However, HEPSO found more functions with the best optimum value among all of the algorithms. Especially for the complex composition functions F24−F29, HEPSO found the best optimum values among all of the variants.

It can also be concluded that HDEMCO performed better than SHADE, MPEDE, SLADE, CSPSO, and HNPPSO for the 10-D problems. For instance, HDEMCO performed better than SHADE for 10 functions and yielded similar performance for 18 functions and inferior performance for two functions.

For the 10-D problems, HDEMCO outperformed the other algorithms for F29 and F30, while it performed poorly for F22 and F23. For F30, HDEMCO outperformed the other algorithms, while for F22, HDEMCO only outperformed SLADE and was surpassed by all of the other algorithms.

Focusing on the different types of functions, the following observations can be made. (1) For the unimodal functions F1–F3, all of the DE variants could find the global optimum, while the PSO variants performed worse than the DE variants. (2) For the simple multimodal functions F4–F10, HDEMCO performed worse than JADE and pbestrrJADE. For example, HDEMCO performed better than pbestrrJADE for one function and yielded similar performance for two functions and inferior performance for four functions. However, HDEMCO outperformed MEPDE, SLADE, CSPAOS, and HNPPSO for these functions. (3) Among the hybrid functions F11–F20, HDEMCO could find the global optimum for five functions (F11, F14, F17, F19, and F20) but was surpassed by JADE and pbestrrJADE again. For instance, HDEMCO performed better than JADE for three functions and yielded similar performance for three functions and inferior performance for four functions. None of the other algorithms, i.e., SHADE, MPEDE, SLADE, HEPSO, CSPSO, HNPPSO, could surpass HDEMCO for these functions. (4) For the composition functions F21–F30, HEPSO outperformed the other algorithms and yielded the best optimum value for eight functions (F21, F22, F24–F29) among all of the variants. HDEMCO performed better than HEPSO for one function and yielded similar performance for one function and inferior performance for eight functions. CSPSO also performed better the HDEMCO for this sub-group, surpassing HDEMCO for seven functions and exhibiting

Table 5
Means and standard deviations of the error for the $D = 30$ problems. The best entries appear in bold.

Functions		JADE	SHADE	MPEDE	SLADE	pbestrrJADE	HEPSO	CSPSO	HNPPSO	HDEMCO
F1	Mean	**0.00E+000**	**0.00E+000**	**0.00E+000**	**0.00E+000**	**0.00E+000**	6.78E+002	5.34E+002	**0.00E+000**	**0.00E+000**
	Std.	**0.00E+000**	**0.00E+000**	**0.00E+000**	**0.00E+000**	**0.00E+000**	8.91E+002	5.74E+002	**0.00E+000**	**0.00E+000**
	+/=/−	=	=	=	=	=	+	+	=	
F2	Mean	**0.00E+000**	**0.00E+000**	**0.00E+000**	2.70E−005	**0.00E+000**	2.36E−006	1.43E−006	6.10E−006	**0.00E+000**
	Std.	**0.00E+000**	**0.00E+000**	**0.00E+000**	7.54E−005	**0.00E+000**	3.36E−006	1.69E−006	2.52E−006	**0.00E+000**
	+/=/−	=	=	=	+	=	+	+	+	
F3	Mean	**0.00E+000**	**0.00E+000**	**0.00E+000**	**0.00E+000**	8.20E−005	2.51E−002	**0.00E+000**	**0.00E+000**	**0.00E+000**
	Std.	**0.00E+000**	**0.00E+000**	**0.00E+000**	**0.00E+000**	1.86E−004	5.43E−002	**0.00E+000**	**0.00E+000**	**0.00E+000**
	+/=/−	=	=	=	=	+	+	=	=	
F4	Mean	3.89E+001	1.37E+001	3.95E+001	3.74E+001	3.09E+001	**1.38E−002**	2.18E+000	3.33E+000	5.55E+000
	Std.	2.91E+001	2.42E+001	2.89E+001	2.97E+001	3.00E+001	**1.89E−002**	2.00E+000	1.48E+000	1.35E+001
	+/=/−	+	+	+	+	+	−	−	−	
F5	Mean	2.32E+001	**1.66E+001**	3.46E+001	4.34E+001	1.88E+001	1.85E+001	2.97E+001	1.08E+002	1.68E+001
	Std.	5.56E+000	**3.52E+000**	8.92E+000	1.30E+001	3.00E+000	4.02E+000	8.90E+000	9.67E+000	3.61E+000
	+/=/−	+	=	+	+	+	+	+	+	
F6	Mean	**0.00E+000**	**0.00E+000**	**0.00E+000**	5.93E−001	**0.00E+000**	**0.00E+000**	**0.00E+000**	**0.00E+000**	**0.00E+000**
	Std.	**0.00E+000**	**0.00E+000**	**0.00E+000**	5.88E−001	**0.00E+000**	**0.00E+000**	**0.00E+000**	**0.00E+000**	**0.00E+000**
	+/=/−	=	=	=	+	=	=	=	=	
F7	Mean	5.25E+001	4.54E+001	6.12E+001	7.90E+001	4.81E+001	4.61E+001	5.26E+001	1.50E+002	**4.50E+001**
	Std.	4.47E+000	2.90E+000	9.28E+000	1.26E+001	3.58E+000	4.31E+000	6.79E+000	7.21E+000	**2.95E+000**
	+/=/−	+	=	+	+	+	=	+	+	
F8	Mean	2.51E+001	1.69E+001	3.58E+001	4.40E+001	2.13E+001	**1.68E+001**	5.65E+001	1.00E+002	1.84E+001
	Std.	5.48E+000	3.31E+000	1.02E+001	1.14E+001	2.94E+000	**3.91E+000**	1.48E+001	2.66E+001	3.34E+000
	+/=/−	+	=	+	+	+	=	+	+	
F9	Mean	1.05E−002	1.75E−003	4.77E−002	4.51E+001	5.26E−003	**0.00E+000**	**0.00E+000**	**0.00E+000**	**0.00E+000**
	Std.	2.91E−002	1.25E−002	1.31E−001	3.55E+001	2.12E−002	**0.00E+000**	**0.00E+000**	**0.00E+000**	**0.00E+000**
	+/=/−	+	+	+	+	+	=	=	=	
F10	Mean	1.37E+003	1.50E+003	2.35E+003	2.00E+003	**1.14E+003**	1.96E+003	3.37E+003	6.19E+003	1.49E+003
	Std.	3.55E+002	2.55E+002	4.95E+002	3.83E+002	**1.80E+002**	1.90E+002	2.21E+002	2.71E+002	2.31E+002
	+/=/−	−	=	+	+	−	+	+	+	
F11	Mean	1.49E+001	1.14E+001	2.84E+001	1.00E+002	8.73E+000	**3.36E+000**	1.09E+001	5.48E+001	6.50E+000
	Std.	1.81E+001	1.89E+001	2.20E+001	5.62E+001	1.22E+001	**2.21E+000**	9.01E+000	4.24E+001	1.15E+001
	+/=/−	+	+	+	+	+	=	+	+	
F12	Mean	1.15E+003	1.23E+003	1.14E+003	1.53E+003	1.14E+003	3.45E+004	7.18E+003	7.29E+003	**1.04E+003**
	Std.	4.08E+002	3.66E+002	4.00E+002	1.49E+003	4.01E+002	5.69E+004	2.43E+003	2.54E+003	**3.77E+002**
	+/=/−	=	+	=	+	=	+	+	+	
F13	Mean	4.65E+001	3.76E+001	6.05E+001	7.85E+002	4.08E+001	4.06E+003	3.83E+003	1.10E+004	**2.99E+001**
	Std.	2.81E+001	2.01E+001	6.90E+001	8.30E+002	1.95E+001	2.43E+003	3.26E+003	5.37E+003	**1.83E+001**
	+/=/−	+	+	+	+	+	+	+	+	
F14	Mean	2.74E+001	3.03E+001	3.29E+001	1.02E+002	3.53E+001	6.46E+002	3.82E+002	1.00E+003	**2.18E+001**
	Std.	1.32E+001	6.41E+000	1.27E+001	6.28E+001	1.30E+001	1.60E+003	3.11E+002	1.49E+003	**9.38E+000**
	+/=/−	+	+	+	+	+	+	+	+	
F15	Mean	3.10E+001	2.09E+001	1.86E+001	2.09E+002	2.62E+001	2.41E+002	1.90E+002	8.88E+002	**1.15E+001**
	Std.	3.03E+001	1.64E+001	2.29E+001	1.38E+002	1.57E+001	5.22E+002	1.67E+002	1.00E+003	**6.26E+000**
	+/=/−	+	+	+	+	+	+	+	+	
F16	Mean	2.89E+002	2.40E+002	5.67E+002	3.27E+002	2.36E+002	3.25E+002	**1.47E+002**	2.25E+002	2.34E+002
	Std.	1.31E+002	1.27E+002	2.43E+002	1.73E+002	1.19E+002	1.06E+002	**9.56E+001**	2.76E+002	1.26E+002
	+/=/−	+	=	+	+	=	+	−	−	
F17	Mean	2.22E+001	3.04E+001	1.03E+002	1.09E+002	3.17E+001	2.56E+001	4.43E+001	8.55E+001	**2.16E+001**
	Std.	2.61E+001	9.59E+000	1.05E+002	1.14E+002	1.31E+001	1.72E+001	7.71E+000	4.73E+001	**1.27E+001**
	+/=/−	=	+	+	+	+	=	+	+	
F18	Mean	5.91E+001	7.41E+001	8.53E+001	1.28E+002	2.30E+002	7.23E+003	2.84E+004	5.67E+001	**3.80E+001**
	Std.	6.63E+001	5.31E+001	8.01E+001	3.75E+002	2.79E+002	9.20E+003	1.00E+004	1.41E+001	**2.49E+001**
	+/=/−	=	+	+	+	+	+	+	+	
F19	Mean	1.43E+001	1.00E+001	2.34E+001	1.25E+002	1.68E+001	5.50E+002	3.50E+002	**4.67E+000**	7.91E+000
	Std.	1.29E+001	3.05E+000	2.65E+001	6.41E+001	1.41E+001	5.81E+002	3.27E+002	**1.84E+000**	2.22E+000
	+/=/−	+	+	+	+	+	+	+	−	
F20	Mean	4.64E+001	2.71E+001	1.44E+002	8.50E+001	3.42E+001	1.07E+002	5.07E+001	1.49E+002	**1.78E+001**
	Std.	5.91E+001	3.55E+001	1.08E+002	7.56E+001	4.22E+001	4.98E+001	2.56E+001	3.37E+001	**3.25E+001**
	+/=/−	=	+	+	+	+	+	+	+	
F21	Mean	2.27E+002	2.19E+002	2.35E+002	2.47E+002	2.09E+002	**1.30E+002**	2.28E+002	3.22E+002	2.20E+002
	Std.	6.98E+000	3.45E+000	8.72E+000	1.10E+001	4.40E+001	**5.16E+001**	7.76E+000	3.32E+001	3.17E+000
	+/=/−	+	=	+	+	−	−	+	+	
F22	Mean	1.33E+002	1.00E+002	1.42E+002	7.36E+002	1.00E+002	**9.90E+001**	1.00E+002	1.00E+002	1.00E+002
	Std.	2.36E+002	0.00E+000	3.03E+002	1.06E+003	0.00E+000	**7.13E+000**	0.00E+000	0.00E+000	0.00E+000
	+/=/−	=	=	=	+	=	=	=	=	

(*continued on next page*)

Table 5 (*continued*)

Functions		JADE	SHADE	MPEDE	SLADE	pbestrrJADE	HEPSO	CSPSO	HNPPSO	HDEMCO
F23	Mean	3.73E+002	3.62E+002	3.90E+002	3.99E+002	3.70E+002	**2.01E+002**	3.70E+002	3.84E+002	3.62E+002
	Std.	6.95E+000	5.19E+000	1.16E+001	1.51E+001	4.13E+000	**1.38E+002**	8.01E+000	6.69E+001	5.54E+000
	+/=/−	+	=	+	+	+	−	+	+	
F24	Mean	4.44E+002	4.34E+002	4.55E+002	4.68E+002	4.44E+002	**1.81E+002**	4.34E+002	5.56E+002	4.35E+002
	Std.	6.63E+000	4.18E+000	1.02E+001	1.12E+001	6.16E+000	**5.79E+001**	4.75E+000	6.39E+001	3.96E+000
	+/=/−	+	=	+	+	+	−	=	+	
F25	Mean	3.87E+002	**3.86E+002**	3.87E+002	3.90E+002	3.87E+002	3.86E+002	3.86E+002	3.87E+002	3.87E+002
	Std.	2.71E−001	**5.68E−001**	2.69E−001	7.47E+000	2.44E−001	1.09E+000	1.97E+000	1.92E+000	3.26E−001
	+/=/−	=	=	=	+	=	=	=	=	
F26	Mean	1.14E+003	1.11E+003	1.36E+003	1.68E+003	1.17E+003	**2.41E+002**	7.80E+002	3.53E+002	1.15E+003
	Std.	1.95E+002	6.97E+001	2.02E+002	2.18E+002	1.38E+002	**5.07E+001**	3.72E+002	2.67E+002	6.65E+001
	+/=/−	−	−	+	+	+	−	−	−	
F27	Mean	5.02E+002	5.04E+002	5.05E+002	5.23E+002	**4.97E+002**	5.87E+002	5.11E+002	5.10E+002	4.98E+002
	Std.	7.96E+000	7.71E+000	7.69E+000	1.57E+001	**5.92E+000**	1.15E+001	4.28E+000	5.56E+000	5.60E+000
	+/=/−	+	+	+	+	=	+	+	+	
F28	Mean	3.36E+002	3.29E+002	3.41E+002	3.74E+002	3.33E+002	**3.00E+002**	**3.00E+002**	3.02E+002	3.12E+002
	Std.	5.50E+001	4.90E+001	5.82E+001	6.80E+001	5.58E+001	**0.00E+000**	**0.00E+000**	1.44E+001	3.48E+001
	+/=/−	+	+	+	+	+	−	−	−	
F29	Mean	4.20E+002	4.20E+002	4.70E+002	5.34E+002	**3.29E+002**	6.19E+002	4.75E+002	4.25E+002	4.18E+002
	Std.	4.06E+001	3.33E+001	6.39E+001	1.27E+002	**4.74E+002**	4.59E+001	2.05E+001	3.93E+001	2.06E+001
	+/=/−	=	+	+	+	−	+	+	+	
F30	Mean	2.12E+003	2.14E+003	2.12E+003	2.34E+003	4.47E+002	7.49E+003	3.20E+003	**3.97E+002**	2.06E+003
	Std.	1.57E+002	1.44E+002	1.79E+002	3.68E+002	6.14E+001	1.41E+003	3.57E+002	**1.27E+001**	1.02E+002
	+/=/−	+	+	=	+	−	+	+	−	
w/t/l	17/11/2	15/14/1	22/8/0	28/2/0	17/9/4	16/8/6	20/6/4	18/6/6		

similar performance for one function and inferior performance for two functions.

4.3.2. Performances for the 30-D problems

The experimental results for the $D = 30$ problems are presented in Table 5. As can be observed, the proposed HDEMCO algorithm provided the best overall performance among all eight variants. Table 7 shows that the lowest rank score in the Friedman test was obtained by HDEMCO (rank = 2.76), with SHADE (rank = 3.58) being the second-best-performing algorithm. According to the experimental results, HDEMCO performed better than SHADE for 15 functions and yielded similar performance for 14 functions and inferior performance for one function. None of the compared algorithms could surpass the proposed HDEMCO for the following functions: F1−F3, F5−F9, F11−F15, F17, F18, F20, F22, F25, and F27. Especially for functions F7, F12−F15, the proposed method achieved the best results among all of the variants.

The following observations can be made by focusing on the different types of functions. (1) For the unimodal functions F1–F3, JADE, SHADE, MEPDE, and HDEMCO could find the global optimum. The other variants yielded worse performance than in the 10-D problems. (2) For the simple multimodal functions F4–F10, HDEMCO outperformed the other variants. It was surpassed by others only in five cases: by HEPSO, CSPSO, and HNNPSO for F4 and by JADE and pbestrrJADE for F10. None of the variants could surpass HDEMCO for any other function. The proposed algorithm performed significantly better than the others particularly for F5, F7, and F8. For F5, it achieved performance similar to that of SHADE and surpassed all of the other variants. For F7 and F8, it yielded performance similar to that of SHADE and HEPSO and surpassed all of the other variants. For F4, HDEMCO surpassed all of the DE variants, although it was surpassed by the PSO variants. (3) For the hybrid functions F11–F20, HDEMCO outperformed the other variants. It was only surpassed by HNPPSO for F16 and F19 and by CSPSO for F16. No variant could surpass HDEMCO for any other function. For this sub-group, HDEMCO outperformed the other algorithms for the following functions: F11, F13–F15, F17, and F18–F20. Especially for F13–F15, the proposed algorithm surpassed all of the other variants. (4) For the composition functions F21–F30, HEPSO performed better than the proposed algorithm, as in the 10-D problems. It surpassed HDEMCO for five functions and yielded similar performance for two functions and inferior performance for three functions. Furthermore, HEPSO could find the best optimum value among all of the variants for six functions (F21–F24, F26, and F28), while HDEMCO failed to find the best optimum value for any function. On the other hand, HDEMCO surpassed other variants besides HEPSO. For instance, it surpassed JADE for six functions and yielded similar performance for three functions and inferior performance for one function. HDEMCO also outperformed other variants for F2, F27, and F29, while it performed worse for F26.

4.3.3. Performances for the 50-D problems

As can be observed from the experimental results for the 50-D problems in Tables 6 and 7, the proposed HDEMCO algorithm provided the best overall performance among all nine variants. The Friedman test experimental results summarized in Table 7 indicate that the average rank of HDEMCO for all functions was 2.95, the best value among all of the variants. Compared to SHADE, which was the second-best algorithm according to the Friedman test, HDEMCO surpassed SHADE for 10 functions and yielded similar performance for 16 functions and inferior performance for four functions. HDEMCO could find the best optimum among all of the compared variants for nine functions (F1–F3, F7, F12, F14, F18, F20, and F29) in the 50-D problems.

The following observations can be made by focusing on the different types of functions. (1) For the unimodal functions F1–F3, none of the variants could surpass HDEMCO. Especially for F2, HDEMCO achieved performance similar to those of JADE and SHADE, surpassing all of the other variants. (2) For the simple multimodal functions F4–F10, HDECMO yielded performance similar to those of JADE, pbestrrJADE, and three PSO variants. For example, HDEMCO surpassed the PSO variants for four functions and produced inferior performance for three functions. For F7, HDEMCO achieved performance similar to that of SHADE, surpassing all of the other variants. On the other hand, for F6, HDEMCO only surpassed SHADE and SHALE and was surpassed by all of the other

Table 6
Means and standard deviations of the error for the $D = 50$ problems. The best entries appear in bold.

Functions		JADE	SHADE	MPEDE	SLADE	pbestrrJADE	HEPSO	CSPSO	HNPPSO	HDEMCO
F1	Mean	**0.00E+000**	**0.00E+000**	**0.00E+000**	**0.00E+000**	**0.00E+000**	1.87E+003	3.01E+002	**0.00E+000**	**0.00E+000**
	Std.	**0.00E+000**	**0.00E+000**	**0.00E+000**	**0.00E+000**	**0.00E+000**	2.28E+003	4.06E+002	**0.00E+000**	**0.00E+000**
	+/=/−	=	=	=	=	=	+	+	+	
F2	Mean	2.21E−007	5.43E−008	9.22E−008	4.41E+014	1.31E−007	4.78E+011	4.95E−006	6.95E−005	**3.92E−008**
	Std.	7.67E−007	1.67E−007	3.75E−007	3.14E+015	3.23E−007	2.88E+012	3.77E−006	3.18E−005	**1.51E−007**
	+/=/−	=	=	+	+	+	+	+	+	
F3	Mean	1.91E−007	**0.00E+000**	**0.00E+000**	**0.00E+000**	4.46E+000	2.26E+003	**0.00E+000**	**0.00E+000**	**0.00E+000**
	Std.	5.95E−007	**0.00E+000**	**0.00E+000**	**0.00E+000**	7.81E+000	4.45E+003	**0.00E+000**	**0.00E+000**	**0.00E+000**
	+/=/−	+	=	=	=	+	+	=	=	
F4	Mean	3.45E+001	4.19E+001	5.10E+001	6.10E+001	3.84E+001	**5.68E−002**	4.01E+000	1.28E+001	2.65E+001
	Std.	4.54E+001	4.90E+001	4.96E+001	4.16E+001	4.79E+001	**6.45E−002**	9.69E+000	2.00E+001	4.18E+001
	+/=/−	=	+	+	+	=	−	−	−	
F5	Mean	4.96E+001	**4.72E+001**	7.56E+001	1.24E+002	5.01E+001	7.73E+001	6.00E+001	3.14E+002	4.80E+001
	Std.	5.90E+000	**7.39E+000**	1.32E+001	3.42E+001	5.63E+000	1.03E+001	1.42E+001	1.44E+001	6.56E+000
	+/=/−	=	=	+	+	=	+	+	+	
F6	Mean	**0.00E+000**	2.87E−006	6.02E−008	7.81E−001	**0.00E+000**	**0.00E+000**	**0.00E+000**	**0.00E+000**	5.13E−007
	Std.	**0.00E+000**	3.33E−006	2.42E−007	1.08E+001	**0.00E+000**	**0.00E+000**	**0.00E+000**	**0.00E+000**	1.72E−006
	+/=/−	−	+	−	+	−	−	−	−	
F7	Mean	9.55E+001	8.90E+001	1.22E+002	2.29E+002	9.72E+001	1.10E+002	9.48E+001	3.39E+002	**8.81E+001**
	Std.	6.51E+000	5.24E+000	1.41E+001	4.31E+001	6.07E+000	1.17E+001	1.13E+001	1.36E+001	**5.65E+000**
	+/=/−	+	=	+	+	+	+	+	+	
F8	Mean	**4.66E+001**	4.68E+001	7.92E+001	1.19E+002	4.90E+001	7.88E+001	6.74E+001	3.09E+002	4.76E+001
	Std.	**7.09E+000**	8.01E+000	1.42E+001	2.76E+001	7.07E+000	1.23E+001	1.58E+001	1.39E+001	6.61E+000
	+/=/−	=	=	+	+	=	+	+	+	
F9	Mean	1.71E+000	8.36E−001	4.71E+000	5.73E+002	1.84E+000	**0.00E+000**	6.60E−002	1.59E−002	9.89E−001
	Std.	1.28E+000	7.73E−001	5.64E+000	2.54E+002	1.47E+000	**0.00E+000**	1.57E−001	6.71E−002	9.52E−001
	+/=/−	+	=	+	+	+	−	−	−	
F10	Mean	**2.66E+003**	3.60E+003	4.82E+003	4.63E+003	2.88E+003	3.98E+003	9.20E+003	1.20E+004	3.28E+003
	Std.	**2.91E+002**	3.86E+002	8.09E+002	6.27E+002	2.72E+002	3.47E+002	2.38E+002	3.18E+002	4.47E+000
	+/=/−	−	+	+	+	−	+	+	+	
F11	Mean	9.64E+001	8.95E+001	1.25E+002	2.00E+002	8.61E+001	1.40E+002	**3.43E+001**	1.46E+002	8.16E+001
	Std.	4.80E+001	2.65E+001	4.82E+001	5.94E+001	4.48E+001	5.81E+001	**7.21E+000**	2.12E+001	3.18E+001
	+/=/−	=	=	+	+	=	+	−	+	
F12	Mean	1.88E+003	1.88E+003	2.08E+003	1.64E+004	1.73E+003	1.13E+006	1.97E+004	2.28E+004	**1.66E+003**
	Std.	4.10E+002	4.79E+002	4.13E+002	2.76E+004	5.45E+002	8.03E+005	8.82E+003	7.29E+003	**4.24E+002**
	+/=/−	+	+	+	+	=	+	+	+	
F13	Mean	3.00E+002	3.20E+002	8.77E+002	1.76E+003	3.44E+002	1.37E+003	**4.51E+001**	1.04E+003	2.28E+002
	Std.	1.53E+002	1.86E+002	6.33E+002	1.25E+003	3.11E+002	9.53E+002	**5.40E+001**	1.30E+003	1.50E+002
	+/=/−	+	+	+	+	+	+	−	+	
F14	Mean	2.31E+002	2.04E+002	2.12E+002	2.86E+002	2.47E+002	6.96E+004	4.09E+003	2.77E+003	**1.41E+002**
	Std.	7.55E+001	5.21E+001	1.12E+002	7.44E+001	1.22E+002	6.87E+004	2.41E+003	8.02E+003	**4.72E+001**
	+/=/−	+	+	+	+	+	+	+	+	
F15	Mean	2.82E+002	2.44E+002	2.39E+002	4.69E+002	2.42E+002	3.30E+003	2.43E+003	**6.23E+001**	1.59E+002
	Std.	1.40E+002	1.08E+002	1.77E+002	1.87E+002	1.20E+002	1.62E+003	2.23E+003	**2.62E+001**	8.81E+001
	+/=/−	+	+	=	+	+	+	+	−	
F16	Mean	6.88E+002	**6.53E+002**	1.17E+003	9.27E+002	7.65E+002	8.11E+002	6.63E+002	1.28E+003	6.91E+002
	Std.	1.95E+002	**1.44E+002**	3.11E+002	2.93E+002	1.53E+002	2.23E+002	1.85E+002	6.59E+002	1.61E+002
	+/=/−	=	=	+	+	+	+	=	+	
F17	Mean	**4.37E+002**	4.47E+002	7.67E+002	6.06E+002	4.76E+002	7.10E+002	5.74E+002	1.10E+003	4.46E+002
	Std.	**1.20E+002**	1.15E+002	2.49E+002	2.17E+002	1.30E+002	1.55E+002	1.24E+002	4.27E+002	8.85E+001
	+/=/−	=	=	+	+	=	+	+	+	
F18	Mean	2.05E+002	6.06E+001	1.07E+002	2.03E+002	6.14E+001	1.81E+005	2.09E+005	1.70E+002	**5.48E+001**
	Std.	4.43E+002	2.84E+001	7.83E+001	4.46E+002	2.03E+001	1.45E+005	1.65E+005	2.15E+001	**1.99E+001**
	+/=/−	+	=	+	+	+	+	+	+	
F19	Mean	1.30E+002	1.08E+002	1.43E+002	1.71E+002	9.44E+001	7.40E+003	8.98E+003	**2.57E+001**	8.55E+001
	Std.	5.05E+001	3.67E+001	6.63E+001	6.91E+001	4.55E+001	2.82E+003	4.35E+003	**1.24E+001**	3.35E+001
	+/=/−	+	+	+	+	=	+	+	−	
F20	Mean	2.46E+002	2.59E+002	6.58E+002	4.07E+002	2.95E+002	4.84E+002	3.86E+002	2.77E+002	**2.44E+002**
	Std.	1.00E+002	1.09E+002	2.71E+002	1.77E+002	1.31E+002	1.40E+002	9.40E+001	3.34E+002	**1.14E+002**
	+/=/−	=	=	+	+	+	+	+	+	
F21	Mean	2.49E+002	**2.42E+002**	2.70E+002	3.18E+002	2.59E+002	2.86E+002	2.56E+002	5.11E+002	2.45E+002
	Std.	7.20E+000	**5.00E+000**	1.55E+001	3.16E+001	7.12E+000	1.31E+001	1.73E+001	1.34E+001	7.08E+000
	+/=/−	+	=	+	+	+	+	+	+	
F22	Mean	2.81E+003	3.63E+003	4.94E+003	4.94E+003	2.10E+003	5.83E+002	**1.00E+002**	**1.00E+002**	2.92E+003
	Std.	1.31E+003	1.5866e+03	1.30E+003	6.42E+002	1.71E+003	1.48E+003	**0.00E+000**	**0.00E+000**	2.04E+003
	+/=/−	−	=	+	+	−	−	−	−	

(*continued on next page*)

Table 6 (*continued*)

Functions		JADE	SHADE	MPEDE	SLADE	pbestrrJADE	HEPSO	CSPSO	HNPPSO	HDEMCO
F23	Mean	4.72E+002	**4.66E+002**	5.04E+002	5.63E+002	4.85E+002	5.27E+002	4.82E+002	7.26E+002	4.65E+002
	Std.	9.30E+000	**1.00E+001**	2.03E+001	3.20E+001	1.02E+001	7.67E+001	1.30E+001	1.37E+001	8.76E+000
	+/=/−	+	=	+	+	+	+	+	+	
F24	Mean	5.42E+002	**5.33E+002**	5.67E+002	6.16E+002	5.65E+002	6.73E+002	5.33E+002	7.97E+002	5.35E+002
	Std.	1.02E+001	**6.87E+000**	1.72E+001	2.84E+001	1.28E+001	1.81E+002	1.03E+001	1.34E+001	9.08E+000
	+/=/−	+	=	+	+	+	+	=	+	
F25	Mean	5.33E+002	5.38E+002	5.27E+002	5.27E+002	5.36E+002	**4.36E+002**	5.30E+002	5.59E+002	5.57E+002
	Std.	3.91E+001	3.78E+001	3.49E+001	4.07E+001	2.65E+001	**1.50E+001**	3.75E+001	1.74E+001	3.28E+001
	+/=/−	−	−	−	−	−	−	−	=	
F26	Mean	1.64E+003	1.54E+003	1.88E+003	2.67E+003	1.80E+003	**3.67E+002**	1.70E+003	3.00E+002	1.61E+003
	Std.	9.01E+001	1.05E+002	1.96E+002	3.72E+002	1.05E+002	**3.49E+002**	1.26E+002	0.00E+00	1.00E+002
	+/=/−	+	−	+	+	+	−	+	−	
F27	Mean	5.73E+002	**5.45E+002**	5.96E+002	6.97E+002	5.86E+002	9.38E+002	6.12E+002	5.50E+002	5.54E+002
	Std.	3.36E+001	**1.58E+001**	5.27E+001	7.27E+001	4.41E+001	3.45E+001	3.44E+001	1.38E+001	2.23E+001
	+/=/−	+	−	+	+	+	+	+	=	
F28	Mean	5.00E+002	**3.28E+002**	4.91E+002	4.94E+002	4.97E+002	4.72E+002	4.81E+002	4.96E+002	4.95E+002
	Std.	2.11E+001	**5.09E+001**	2.02E+001	1.74E+001	1.62E+001	4.76E+000	1.89E+001	1.05E+001	1.41E+001
	+/=/−	=	−	=	=	=	−	−	=	
F29	Mean	3.83E+002	3.92E+002	5.89E+002	9.98E+002	4.41E+002	8.55E+002	5.76E+002	5.74E+002	**3.70E+002**
	Std.	5.66E+001	6.69E+001	1.64E+002	3.93E+002	6.09E+001	1.20E+002	1.01E+002	2.30E+002	**5.54E+001**
	+/=/−	=	+	+	+	+	+	+	+	
F30	Mean	6.59E+005	6.25E+005	7.35E+005	7.87E+005	**1.03E+003**	5.28E+006	7.67E+005	1.09E+003	6.03E+005
	Std.	8.51E+004	4.29E+004	1.42E+005	1.38E+005	**1.54E+002**	5.59E+005	4.18E+004	4.87E+001	3.37E+004
	+/=/−	+	+	+	+	−	+	+	−	
w/t/l	15/11/4	10/16/4	24/4/2	26/3/1	16/9/5	23/0/7	19/3/8	18/4/8		

Table 7
Average ranking of competitor algorithms for the $D=10$, $D=30$, and $D=50$ problems, as obtained from the Friedman test.

	JADE	SHADE	MPEDE	SLADE	pbestrrJADE	HEPSO	CSPSO	HNPPSO	HDEMCO	*p*-value
$D=10$	4.00	4.45	6.00	7.60	3.58	**3.52**	5.40	6.12	3.83	3.38E−014
$D=30$	4.68	3.58	6.13	7.50	4.18	4.98	5.38	5.78	**2.76**	5.01E−012
$D=50$	4.18	3.37	5.98	7.21	4.45	6.42	4.86	5.58	**2.95**	9.87E−012

variants. (3) For the hybrid functions F11–F20, the proposed algorithm outperformed the other variants and was surpassed only in four cases: by CSPSO for F11 and F13 and by HNPPSO for F15 and F19. Besides CSPSO and HNPPSO, none of the other variants could surpass HDEMCO for this sub-group. HDEMCO also achieved better performance than all of the other variants for F12, F14, F16, F18, and F20. For instance, for F14, HDEMCO surpassed all of the other variants. (4) For the composition functions F21–F30, HDEMCO achieved better performance than all of the other variants besides SHADE. It was surpassed by SHADE for four functions and yielded similar performance for four functions and better performance for two functions. HDEMCO also achieved superior performance for F21, F23, F24, and F29. For instance, for F21 and F23, HDEMCO performed similarly to SHADE and surpassed all of the other variants. For other functions, such as F22 and F25, the proposed algorithm yielded performance inferior to those of other variants. For instance, for F25, HDEMCO performed similarly to JADE and was surpassed by other algorithms.

By simultaneously considering the 10-D, 30-D, and 50-D problems, the following conclusions can be drawn for the proposed algorithm:

(1) It can be observed that for the 30-D and 50-D problems, the HDEMCO algorithm performed better than the other variants than for the 10-D problems. The average rankings of HDEMCO obtained by the Friedman test for 10-D, 30-D, and 50-D problems were 3.83 (the third-best value), 2.76 (the best value), and 2.95 (the best value) respectively. The same conclusion can be drawn for SHADE. It achieved 4.45 (the fifth-best value), 3.58 (the second-best value), and 3.47 (the second-best value) for the 10-D, 30-D, and 50-D problems, respectively. In the proposed algorithm, the bottom layer is implemented by SHADE with a modified parameter *p*, which may be why HDEMCO and SHADE presented similar features for the different dimensions.

(2) For the unimodal functions F1–F3, the proposed HDEMCO algorithm provided the best overall performance among all nine variants. HDEMCO could find the global optimum, except for F2 in the 50-D problem. The top layer in the proposed algorithm is implemented by MCO, which is characterized by a high convergence speed and good local search capability. These are also advantages of the proposed algorithm.

(3) Another advantage of the proposed algorithm is it performs better for solving 30-D problems than the other variants. In 30-D problems, only for the composition functions F21–F30, HEPSO performed better than the proposed algorithm. No other variants can surpass the HDEMCO for other types of functions. The average rankings of HDEMCO obtained by the Friedman test for 30-D and 50-D problems were both the best values. Compared to SHADE, which provides the second-best values for 30-D and 50-D problems, HDEMCO surpassed SHADE for 15 and 10 functions for 30-D and 50-D problems, respectively. In other words, HDEMCO's superior performance compared to SHADE weakens with the increasing dimension. Therefore, for the real optimization problems with 30-D in expert systems, the proposed algorithm is a potential choice.

4.4. Comparison with other SI algorithms

The experimental results for $D=30$ problems are shown in Table 8. This table provides the mean values and standard deviations obtained using the HDEMCO and other SI algorithms considered herein for all 30 benchmark functions. The experimental

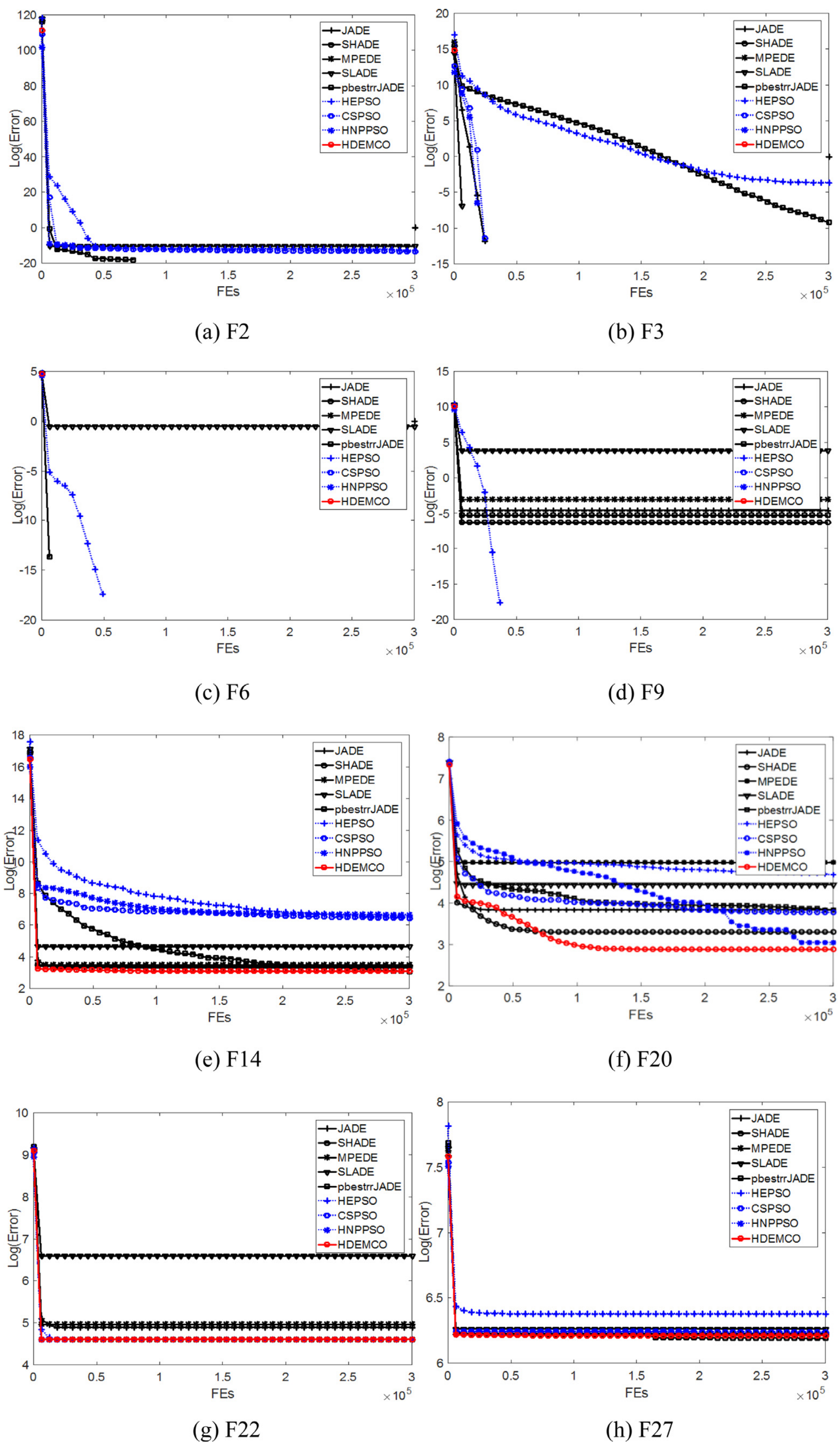


Fig. 2. Mean convergence curves of the benchmark functions: unimodal (F2, F3), simple multimodal (F6, F9), hybrid (F14, F20), and composition (F22, F27) functions.

Table 8
Means and standard deviations of the error for the $D = 30$ problems compared to other SI algorithms. The best entries appear in bold.

Functions		GA	DE	PSO	ACO	CS	HDEMCO
F1	Mean	**0.00E+000**	4.10E+003	7.79E+002	1.10E+001	1.66E+003	**0.00E+000**
	Std.	**0.00E+000**	2.05E+002	8.72E+002	5.81E+000	8.45E+002	**0.00E+000**
	+/=/−	=	+	+	+	+	
F2	Mean	**0.00E+000**	1.90E−005	4.34E+003	4.32E+002	2.62E+002	**0.00E+000**
	Std.	**0.00E+000**	6.64E−005	1.86E+002	5.80E+001	4.05E+001	**0.00E+000**
	+/=/−	=	+	+	+	+	
F3	Mean	**0.00E+000**	**0.00E+000**	**0.00E+000**	7.27E+002	**0.00E+000**	**0.00E+000**
	Std.	**0.00E+000**	**0.00E+000**	**0.00E+000**	8.20E+001	**0.00E+000**	**0.00E+000**
	+/=/−	=	=	=	+	=	
F4	Mean	5.43E+001	3.36E+001	7.52E+001	**9.18E−001**	7.93E+000	5.55E+000
	Std.	1.64E+001	3.34E+001	5.41E+001	**3.22E−001**	1.36E+001	1.35E+001
	+/=/−	+	+	+	−	+	
F5	Mean	2.52E+001	1.36E+002	5.28E+001	5.58E+001	1.21E+002	**1.68E+001**
	Std.	6.30E+000	3.14E+001	1.02E+001	4.95E+000	1.42E+001	**3.61E+000**
	+/=/−	+	+	+	+	+	
F6	Mean	**0.00E+000**	3.20E+001	**0.00E+000**	**0.00E+000**	8.53E+000	**0.00E+000**
	Std.	**0.00E+000**	9.71E+000	**0.00E+000**	**0.00E+000**	2.25E+000	**0.00E+000**
	+/=/−	=	+	=	=	+	
F7	Mean	5.88E+001	2.56E+002	9.00E+001	7.98E+001	1.83E+002	**4.50E+001**
	Std.	8.19E+000	7.68E+001	1.52E+001	5.24E+000	2.16E+001	**2.95E+000**
	+/=/−	+	+	+	+	+	
F8	Mean	2.60E+001	1.20E+002	5.50E+001	6.10E+001	1.23E+002	**1.84E+001**
	Std.	8.31E+000	3.27E+001	1.46E+001	6.06E+000	1.85E+001	**3.34E+000**
	+/=/−	+	+	+	+	+	+
F9	Mean	1.81E−002	2.19E+003	8.01E+000	2.87E+002	4.78E+003	**0.00E+000**
	Std.	9.08E−002	1.33E+003	8.09E+000	9.74E+001	9.44E+002	**0.00E+000**
	+/=/−	+	+	+	+	+	
F10	Mean	2.58E+003	3.63E+003	2.20E+003	**1.44E+003**	2.68E+003	1.49E+003
	Std.	6.33E+002	6.77E+002	5.05E+002	**1.50E+002**	2.29E+002	2.31E+002
	+/=/−	+	+	+	−	+	
F11	Mean	1.77E+001	1.93E+002	4.22E+001	1.45E+002	4.93E+001	**6.50E+000**
	Std.	2.19E+001	5.49E+001	2.58E+001	3.51E+001	1.24E+001	**1.15E+001**
	+/=/−	+	+	+	+	+	
F12	Mean	2.41E+003	6.97E+003	3.66E+004	2.44E+005	3.50E+005	**1.04E+003**
	Std.	5.03E+003	1.50E+004	9.30E+004	1.01E+005	1.30E+005	**3.77E+002**
	+/=/−	+	+	+	+	+	
F13	Mean	**2.87E+001**	6.91E+003	1.23E+004	5.54E+003	2.95E+004	2.99E+001
	Std.	**1.12E+001**	1.95E+004	1.13E+004	1.61E+003	5.78E+003	1.83E+001
	+/=/−	=	+	+	+	+	
F14	Mean	**2.14E+001**	2.11E+002	1.46E+002	2.08E+004	6.60E+002	2.18E+001
	Std.	**1.14E+001**	6.92E+001	6.07E+001	8.53E+003	3.49E+002	9.38E+000
	+/=/−	=	+	+	+	+	
F15	Mean	**5.68E+000**	4.32E+002	1.11E+004	6.23E+002	9.62E+002	1.15E+001
	Std.	**3.90E+000**	1.00E+003	8.60E+003	2.30E+002	1.64E+002	6.26E+000
	+/=/−	−	+	+	+	+	
F16	Mean	6.83E+002	9.82E+002	5.65E+002	2.66E+002	3.40E+002	**2.34E+002**
	Std.	3.92E+002	3.09E+002	2.02E+002	8.90E+001	1.43E+002	**1.26E+002**
	+/=/−	+	+	+	=	=	
F17	Mean	1.67E+002	6.31E+002	7.19E+001	7.94E+001	1.09E+002	**2.16E+001**
	Std.	1.36E+002	2.76E+002	4.89E+001	1.13E+001	2.39E+001	**1.27E+001**
	+/=/−	+	+	+	+	+	
F18	Mean	**2.32E+001**	2.71E+003	5.97E+003	8.42E+004	4.37E+004	3.80E+001
	Std.	**8.76E+000**	1.29E+004	6.89E+003	2.10E+004	9.34E+003	2.49E+001
	+/=/−	−	+	+	+	+	
F19	Mean	**4.10E+000**	1.15E+003	8.13E+003	2.02E+003	3.90E+002	7.91E+000
	Std.	**1.46E+000**	3.57E+003	9.65E+003	1.02E+003	1.15E+002	2.22E+000
	+/=/−	−	+	+	+	+	
F20	Mean	1.05E+002	5.93E+002	1.05E+002	9.06E+001	1.31E+002	**1.78E+001**
	Std.	1.16E+002	1.79E+002	5.88E+001	1.98E+001	3.44E+001	**3.25E+001**
	+/=/−	+	+	+	+	+	
F21	Mean	2.27E+002	3.40E+002	2.56E+002	**1.17E+002**	3.02E+002	2.20E+002
	Std.	1.00E+001	3.44E+001	1.39E+001	**2.80E+000**	6.82E+001	3.17E+000
	+/=/−	+	+	+	=	+	
F22	Mean	7.84E+002	3.65E+003	1.01E+002	1.12E+002	2.70E+002	**1.00E+002**
	Std.	1.40E+003	1.64E+003	1.32E+000	2.24E+000	5.31E+002	**0.00E+000**
	+/=/−	+	+	=	+	+	

(*continued on next page*)

Table 8 (*continued*)

Functions		GA	DE	PSO	ACO	CS	HDEMCO
F23	Mean	3.75E+002	5.42E+002	4.18E+002	3.87E+002	4.53E+002	**3.62E+002**
	Std.	1.17E+001	5.91E+001	2.53E+001	3.81E+000	2.36E+001	**5.54E+000**
	+/=/−	+	+	+	+	+	
F24	Mean	4.53E+002	6.04E+002	4.84E+002	**1.21E+002**	4.95E+002	4.35E+002
	Std.	1.27E+001	5.95E+001	2.54E+001	**8.46E+000**	1.24E+002	3.96E+000
	+/=/−	+	+	+	−	+	
F25	Mean	3.86E+002	3.91E+002	4.01E+002	**3.83E+002**	3.83E+002	3.87E+002
	Std.	8.41E−002	1.29E+001	1.48E+001	**2.96E−002**	8.21E−002	3.26E−001
	+/=/−	=	=	+	=	=	
F26	Mean	1.26E+003	2.94E+003	1.56E+003	**2.53E+002**	4.17E+002	1.15E+003
	Std.	1.80E+002	7.14E+002	6.21E+002	**6.97E+000**	1.81E+002	6.65E+001
	+/=/−	=	+	+	−	+	
F27	Mean	4.99E+002	5.57E+002	5.56E+002	**4.98E+002**	5.09E+002	4.98E+002
	Std.	7.21E+000	4.38E+001	2.15E+001	**4.20E+000**	6.60E+000	5.60E+000
	+/=/−	=	+	+	=	+	
F28	Mean	3.38E+002	3.85E+002	3.83E+002	**3.08E+002**	3.40E+002	3.12E+002
	Std.	5.84E+001	7.92E+001	6.08E+001	**6.05E+000**	3.65E+001	3.48E+001
	+/=/−	+	+	+	−	+	
F29	Mean	4.72E+002	1.23E+003	5.20E+002	4.71E+002	5.65E+002	**4.18E+002**
	Std.	7.25E+001	2.71E+002	8.02E+001	2.70E+001	5.50E+001	**2.06E+001**
	+/=/−	+	+	+	+	+	
F30	Mean	**2.02E+003**	2.57E+003	7.81E+003	7.35E+003	2.50E+004	2.06E+003
	Std.	**6.96E+001**	8.86E+002	4.10E+003	1.25E+003	4.74E+003	1.02E+002
	+/=/−	−	+	+	+	+	
w/t/l	17/9/4	28/2/0	27/3/0	20/5/5	27/3/0		
Friedman	2.78	5.12	4.23	2.92	4.23	**1.74**	

results obtained from the Friedman test are shown in the last row of the table.

Clearly, the proposed HDEMCO algorithm provided the best overall performance among all six SI algorithms. Table 8 shows that the lowest rank score in the Friedman test was obtained by HDEMCO (rank = 1.74), with GA (rank = 2.78) being the second-best-performing algorithm. According to the experimental results, HDEMCO performed better than GA for 17 functions and yielded similar performance for nine functions and inferior performance for four functions. None of the compared algorithms could surpass the proposed HDEMCO for the following functions: F1−F3, F5−F9, F11−F14, F16, F17, F20−F23, F25, F27, and F29.

The following observations can be made by focusing on the different types of functions. (1) For the unimodal functions F1–F3, GA and HDEMCO could find the global optimum. (2) For the simple multimodal functions F4–F10, HDEMCO outperformed the other variants. It was only surpassed by ACO for F4 and F10. None of the variants could surpass HDEMCO for any other function. The proposed algorithm also performed significantly better than the others, particularly for F5 and F7–F9. (3) For the hybrid functions F11–F20, HDEMCO outperformed the other variants. It was only surpassed by GA for F15, F18, and F19. No variant could surpass HDEMCO for any other function. For this sub-group, HDEMCO outperformed the other algorithms for the following functions: F11–F14, F16, F17, and F20. (4) For the composition functions F21–F30, the proposed algorithm outperformed the other variants and was surpassed only in four cases: by ACO for F24, F26, and F28 and by GA for F30. HDEMCO also performed better than all of the other variants for F21–F23, F25, F27, and F29.

4.5. Convergence speed comparison

This section presents an analysis of the convergence characteristics of the proposed HDEMCO algorithm and the other compared variants for some selected benchmark functions for a 30-D case. All types of functions were selected, including unimodal (F2, F3), simple multimodal (F6, F9), hybrid (F14, F20), and composition (F22, F27) functions. Fig. 2 presents the convergence graph, showing the logarithmic differences between the optimum solution and the solutions obtained by the algorithm variants versus the number of fitness evaluations (FEs). The logarithmic difference was negative infinity (not shown in the figures) when the difference between the optimum solution and obtained solution was zero. Therefore, some curves stop after a certain number of FEs.

For unimodal functions F2 and F3, the average numbers of FEs necessary for HDEMCO to obtain the global optimum were 1363 and 1653, respectively, which are the second-best values. The fastest convergence was achieved by SHADE, which required 954 and 1261 FEs to obtain the global optimum for F2 and F3, respectively.

For simple multimodal function F6, HDEMCO achieved the third-fastest convergence according to the experimental results. The average number of FEs required by HDEMCO was 1025 for function F6. The best and second-best values were obtained by HNPPSO (FEs = 716) and SHADE (FEs = 875), respectively. For function F9, HDEMCO and the PSO variants could find the global optimum, and HDEMCO achieved the fastest convergence. The average number of FEs required by HDEMCO was 503. The second-best value was achieved by HNPPSO (FEs = 544).

For hybrid (F14, F20) and composition functions (F22), none of the variants could find the global optimum. However, HDEMCO achieved not only a higher solution accuracy than the other variants, but also fast convergence. For composition function F27, HDEMCO achieved the second-best solution accuracy among all of the variants, as well as fast convergence.

In summary, although HDEMCO did not provide the fastest convergence for all of the functions, it ranked in the top three for the unimodal and simple multimodal functions and exhibited fast convergence for the other functions.

5. Conclusion

To balance the global and local search abilities for DE, a hierarchical heterogeneous DE algorithm that incorporates MMCO was proposed in this paper. In the proposed structure, the top layer performs MCO, and the bottom layer conducts SHADE. The MCO search is based on SHADE, but its search results do not affect the bottom layer. Thus, even if the MCO falls into a local optimum, it will not lead to premature completion of the whole algorithm, since the MCO results do not affect SHADE. Therefore, this algorithm can balance the global and local searches. To make the SHADE algorithm more focused on global searching, p is set to 0.5. For the bottom layer, we developed MMCO, which includes a velocity vector, similarly to a PSO algorithm. The first-order stabilities were also analyzed without the stagnation assumption for *gbest* and *pbest*. According to the analysis, when certain conditions are met, the particles in MMCO exhibit first-order stability, and their expected positions converge to u_g–u_x after a long run. Benchmark functions were collected from CEC2017 for testing, and HDEMCO was compared with eight advanced DE and PSO variants to evaluate its effectiveness. The solution accuracy and convergence speed were separately compared during the experiments. The results show that HDEMCO achieved the best performance among all of the variants for most of the benchmark functions, except for the 10-D problems. For example, for the 30-D and 50-D problems, the average ranks obtained from the Friedman test for the proposed method were 2.76 and 2.95, respectively, which were the best values among all of the variants studied. The convergence characteristic comparison results indicate that, although HDEMCO did not provide the fastest convergence for all of the functions, it ranked in the top three for the unimodal and simple multimodal functions and exhibited fast convergence for the other functions.

However, HDEMCO still has some limitations.

(1) Individuals can still move to infinity, even though their expected positions are convergent. Hence, second-order stability analysis is necessary for the proposed MMCO. Unfortunately, second-order stability analysis without the stagnation assumption for PSO is very complex and difficult (Bonyadi & Michalewicz, 2016). Therefore, this issue should be addressed in the future.
(2) For future work, employing HDEMCO in a real application in expert systems would be valuable. Since the proposed algorithm implements the bottom layer using SHADE, the parameters in SHADE are also reserved in the proposed algorithm. On the other hand, a parameter w is introduced in the MMCO for the top layer. How to tune these parameters for real optimization problems in expert systems should be studied carefully in the future.
(3) The top layer in the proposed algorithm is implemented by MCO, which characterized by a high convergence speed and good local search capability. These are also advantages of the proposed algorithm. Combining the MCO into other SI variants, e.g., GA, ACO, and ABC, will be another topic for future research.
(4) According to our experimental results, the superior performance achieved by the proposed algorithm is weakened with the increase in dimension. Therefore, how to increase the performance of the HDEMCO for high-dimensional problems should be addressed in the future.

Conflict of interest

The authors declared that they do not have any commercial or associative interest that represents a conflict of interest in connection with the work submitted.

Credit authorship contribution statement

Zhao-Guang Liu: Conceptualization, Formal analysis, Methodology, Writing - original draft, Writing - review & editing. **Xiu-Hua Ji:** Conceptualization, Formal analysis, Methodology, Writing - original draft, Writing - review & editing. **Yang Yang:** Conceptualization, Software, Writing - original draft, Writing - review & editing, Funding acquisition.

Acknowledgments

This work was supported by the Shandong Provincial Natural Science Foundation, China (Grant nos. ZR2017MF067 and 2016GGX101022).

Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.eswa.2019.04.040.

References

Awad, N. H., Ali, M. Z., Liang, J. J., Qu, B. Y., & Suganthan, P. N. (2017). *Problem definitions and evaluation criteria for the CEC 2017 special session and competition on single objective real-parameter numerical optimization*. Nanyang Technological University, Jordan University of Science and Technology, and Zhengzhou University Technical report.

Bonyadi, M. R., & Michalewicz, Z. (2016). Stability analysis of the particle swarm optimization without stagnation assumption. *IEEE Transactions on Evolutionary Computation, 20*(5), 814–819.

Bonyadi, M. R., & Michalewicz, Z. (2017). Impacts of coefficients on movement patterns in the particle swarm optimization algorithm. *IEEE Transactions on Evolutionary Computation, 21*, 378–389.

Cai, Y., Liao, J., Wang, T., Chen, Y., & Tian, H. (2018). Social learning differential evolution. *Information Sciences, 433–434*, 464–509.

Chen, Z., Zhou, S., & Luo, J. (2017). A robust ant colony optimization for continuous functions. *Expert Systems with Applications, 81*, 309–320.

Chang, W. D. (2007). A multi-crossover genetic approach to multivariable PID controllers tuning. *Expert Systems with Applications, 33*(3), 620–626.

Chang, Y. P. (2010). An ant direction hybrid differential evolution algorithm in determining the tilt angle for photovoltaic modules. *Expert Systems with Applications, 37*(7), 5415–5422.

Coelho, L. D. S., & Bernert, D. L. D. A. (2010). A modified ant colony optimization algorithm based on differential evolution for chaotic synchronization. *Expert Systems with Applications, 37*(6), 4198–4203.

Das, S., Mullick, S. S., & Suganthan, P. N. (2016). Recent advances in differential evolution – An updated survey. *Swarm and Evolutionary Computation, 27*, 1–30.

Derrac, J., García, S., Molina, D., & Herrera, F. (2011). A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation, 1*(1), 3–18.

Dhahri, H., Alimi, A. M., & Abraham, A. (2012). Hierarchical multi-dimensional differential evolution for the design of beta basis function neural network. *Neurocomputing, 97*, 131–140.

Epitropakis, M. G., Plagianakos, V. P., & Vrahatis, M. N. (2012). Evolving cognitive and social experience in particle swarm optimization through differential evolution: A hybrid approach. *Information Sciences, 216*, 50–92.

Escario, J. B., Jimenez, J. F., & Giron-Sierra, J. M. (2012). Optimisation of autonomous ship manoeuvres applying Ant Colony Optimisation metaheuristic. *Expert Systems with Applications, 39*(11), 10120–10139.

Ganapathy, K., Vaidehi, V., Kannan, B., & Murugan, H. (2014). Hierarchical particle swarm optimization with ortho-cyclic circles. *Expert Systems with Applications, 41*(7), 3460–3476.

Gumaida, B. F., & Luo, J. (2017). An efficient algorithm for wireless sensor network localization based on hierarchical structure poly-particle swarm optimization. *Wireless Personal Communications, 97*(1), 125–151.

Hu, C., Liu, H., & Zhang, P. (2016). Cooperative co-evolutionary artificial bee colony algorithm based on hierarchical communication model. *Chinese Journal of Electronics, 25*(3), 570–576.

Kennedy, J., & Eberhart, R. C. (1995). Particle swarm optimization. In *Proceedings of IEEE International conference on neural network* (pp. 1942–1948).

Kennedy, J. (1999). Small worlds and mega-minds: Effects of neighborhood topology on particle swarm performance. In *Proceedings of the congress on evolutionary computation* (pp. 1391–1938).

Liang, J. J., Qin, A. K., Suganthan, P. N., & Baskar, S. (2006). Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. *IEEE Transactions on Evolutionary Computation, 10*(3), 281–295.

Liang, J. J., Qu, B. Y., Suganthan, P. N., & Hernández-Díaz, A. G. (2013). *Problem definitions and evaluation criteria for the CEC 2013 special session on real-parameter optimization*. Zhengzhou University and Nanyang Technological University *Technical report*.

Liao, T., Stützle, T., Ocac, M. A. M. D., & Dorigo, M. (2014). A unified ant colony optimization algorithm for continuous optimization. *European Journal of Operational Research, 234*(3), 597–609.

Lim, W. H., & Isa, N. A. M. (2014). An adaptive two-layer particle swarm optimization with elitist learning strategy. *Information Sciences, 273*, 49–72.

Lin, G. H., Zhang, J., & Liu, Z. H. (2018). Hybrid particle swarm optimization with differential evolution for numerical and engineering optimization. *International Journal of Automation and Computing, 1*, 103–114.

Liu, Y., & Sun, F. (2011). A fast differential evolution algorithm using k-nearest neighbour predictor. *Expert Systems with Applications, 38*(4), 4254–4258.

Liu, Z. G., Ji, X. H., & Liu, Y. X. (2018). Hybrid non-parametric particle swarm optimization and its stability analysis. *Expert Systems with Applications, 92*, 256–275.

Ma, L., Chen, H., Hu, K., & Zhu, Y. (2014). Hierarchical artificial bee colony algorithm for RFID network planning optimization. *The Scientific World Journal*, 941532 21 pages.

Mahmoodabadi, M. J., Mottaghi, Z. S., & Bagheri, A. (2014). HEPSO: High exploration particle swarm optimization. *Information Sciences, 273*, 101–111.

Maio, F. D., Baronchelli, S., & Zio, E. (2014). Hierarchical differential evolution for minimal cut sets identification: Application to nuclear safety systems. *European Journal of Operational Research, 238*(2), 645–652.

María, L. T., Paola, C. S., Germán, B., & Miguel, M. G. (2017). Hierarchical parallel model for improving performance on differential evolution. *Concurrency & Computation Practice & Experience, 401*(10), E4087.

Marinakis, Y., Migdalas, A., & Sifaleras, A. (2017). A hybrid particle swarm optimization – variable neighborhood search algorithm for constrained shortest path problems. *European Journal of Operational Research., 261*(3), 819–834.

Meng, A., Li, Z., Yin, H., Chen, S., & Guo, Z. (2016). Accelerating particle swarm optimization using crisscross search. *Information Sciences, 329*, 52–72.

Mlakar, U., Potocnik, B., & Brest, J. (2016). A hybrid differential evolution for optimal multilevel image thresholding. *Expert Systems with Applications, 65*, 221–232.

Mohamed, A. W., & Suganthan, P. N. (2017). Real-parameter unconstrained optimization based on enhanced fitness-adaptive differential evolution algorithm with novel mutation. *Soft Computing, 22*(10), 3215–3235.

Nazarahari, M., Khanmirza, E., & Doostie, S. (2019). Multi-objective multi-robot path planning in continuous environment using an enhanced genetic algorithm. *Expert Systems with Applications, 115*, 106–120.

Pan, Q. K. (2016). An effective co-evolutionary artificial bee colony algorithm for steelmaking-continuous casting scheduling. *European Journal of Operational Research., 250*(3), 072–714.

Peng, Y., & Lu, B. L. (2013). A hierarchical particle swarm optimizer with latin sampling based memetic algorithm for numerical optimization. *Applied Soft Computing, 13*(5), 2823–2836.

Plessis, M. C., & Engelbrecht, A. P. (2012). Using competitive population evaluation in a differential evolution algorithm for dynamic environments. *European Journal of Operational Research, 218*(1), 7–20.

Rastegar, S., Araújo, R., & Mendes, J. (2017). Online identification of Takagi–Sugeno fuzzy models based on self-adaptive hierarchical particle swarm optimization algorithm. *Applied Mathematical Modelling, 45*, 606–620.

Rusin, M., & Zaitseva, E. (2012). Hierarchical heterogeneous ant colony optimization. In *Proceedings of the IEEE federated conference on computer science and information systems* (pp. 197–203).

Sethanan, K., & Neungmatch, W. (2016). Multi-objective particle swarm optimization for mechanical harvester route planning of sugarcane field operations. *European Journal of Operational Research, 252*(3), 969–984.

Sethanan, K., & Pitakaso, R. (2016). Improved differential evolution algorithms for solving generalized assignment problem. *Expert Systems with Applications, 45*, 450–459.

Shi, Y., & Eberhart, R. C. (1998). Parameter selection in particle swarm optimization. In *Proceedings of the seventh conference on evolutionary programming* (pp. 591–600).

Sreeja, N. K., & Sankar, A. (2016). A hierarchical heterogeneous ant colony optimization based approach for efficient action rule mining. *Swarm and Evolutionary Computation., 29*, 1–2.

Storn, R., & Price, K. (1995). *Differential evolution: A simple and efficient adaptive scheme for global optimisation over continuous spaces*. Berkley: International Computer Science Institute Technical Report TR-95-012.

Storn, R., & Price, K. (1997). Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization, 11*, 341–359.

Tanabe, R., & Fukunaga, A. (2013). Success-history based parameter adaptation for differential evolution. In *Proceedings of the IEEE congress on evolutionary computation* (pp. 71–78).

Tseng, H-E., Chang, C-C., Lee, S-C., & Huang, Y-M. (2018). A block-based genetic algorithm for disassembly sequence planning. *Expert Systems with Applications, 96*, 492–505.

Wang, L., Yang, B., & Chen, Y. (2014). Improving particle swarm optimization using multi-layer searching strategy. *Information Sciences, 274*, 70–94.

Wang, S. C., & Yeh, M-F. (2014). A modified particle swarm optimization for aggregate production planning. *Expert Systems with Applications, 41*(6), 3069–3077.

Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics Bulletin, 1*, 80–83.

Wu, G., Mallipeddi, R., Suganthan, P. N., Wang, R., & Chen, H. (2016). Differential evolution with multi-population based ensemble of mutation strategies. *Information Sciences, 329*, 329–345.

Yang, X-S., & Deb, S. (2010). Engineering optimization by cuckoo search. *International Journal of Mathematical Modelling and Numerical Optimisation, 1*, 330–343.

Yi, W., Zhou, Y., Gao, L., Li, X., & Mou, J. (2016). An improved adaptive differential evolution algorithm for continuous optimization. *Expert Systems with Applications, 44*, 1–12.

Zhang, J., & Sanderson, A. C. (2009). JADE: Adaptive differential evolution with optional external archive. *IEEE Transactions on Evolutionary Computation, 13*(5), 945–958.

Zhang, Z., Zhang, N., & Feng, Z. (2014). Multi-satellite control resource scheduling based on ant colony optimization. *Expert Systems with Applications, 41*, 2816–2823.

Zhao, Z., Yang, J., Hu, Z., & Che, H. (2016). A differential evolution algorithm with self-adaptive strategy and control parameters based on symmetric Latin hypercube design for unconstrained optimization problems. *European Journal of Operational Research, 250*(1), 30–45.

Zhong, J. H., Shen, M., Zhang, J., Chung, H. S. H., Shi, Y. H., & Li, Y. (2013). A differential evolution algorithm with dual populations for solving periodic railway timetable scheduling problem. *IEEE Transactions on Evolutionary Computation, 17*(4), 512–527.

Zorarpacı, E., & Özel, S. A. (2016). A hybrid approach of differential evolution and artificial bee colony for feature selection. *Expert Systems with Applications., 62*, 91–103.