Neurocomputing 221 (2017) 123–137

Contents lists available at ScienceDirect

Neurocomputing

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

Collective decision optimization algorithm: A new heuristic optimization method

Qingyang Zhang, Ronggui Wang, Juan Yang*, Kai Ding, Yongfu Li, Jiangen Hu

School of Computer and Information, Hefei University of Technology, Hefei 230009, China

ARTICLE INFO

Communicated by Jiahu Qin

Keywords:
Collective decision optimization algorithm
Artificial neural networks
Meta-heuristic
Decision-making

ABSTRACT

Recently, inspired by nature, diversiform successful and effective optimization methods have been proposed for solving many complex and challenging applications in different domains. This paper proposes a new meta-heuristic technique, collective decision optimization algorithm (CDOA), for training artificial neural networks. It simulates the social behavior of human based on their decision-making characteristics including experience-based phase, others'-based phase, group thinking-based phase, leader-based phase and innovation-based phase. Different corresponding operators are designed in the methodology. Experimental results carried out on a comprehensive set of benchmark functions and two nonlinear function approximation examples demonstrate that CDOA is competitive with respect to other state-of-art optimization algorithms.

1. Introduction

Global optimization refers to the process of obtaining optimal values for a given system or mathematical model from all the possible solutions to maximize or minimize objective function. Increasing complexity of optimization tasks involved in different fields of real world make the development of optimization techniques more significant and interesting than before. Therefore, over the past two decades, various optimization methods have been proposed on diverse inspirations. According to the number of candidate solutions, they can be grouped in two categories: individual-based and population-based methods. In the former case, optimization process tends to start with single random solution, which is improved over the course iterations. Thus, individual-based techniques need less computation cost and function evaluation but suffer from great drawbacks: derivation-based mechanism and premature convergence. In contrary, in the latter case, a set of solutions is generated randomly and improved from generation to generation. In this way, population-based methods have high ability to avoid local optima, since the exchange of information occurs between the solutions and assists them to conquer different difficulties of search spaces. Meanwhile, they also encounter high computational cost and more function evaluation.

The important feature of population-based stochastic search techniques is the division of the solution domain to two main milestones: diversification and intensification [1]. The former refers to the phase where candidate solutions tend to be changed more frequently and explore promising regions as broad as possible. Contradictory, the latter promotes convergence toward the best values obtained in exploration process. In other words, favoring diversification turns out higher local optima avoidance, whereas emphasizing intensification yields to faster convergence rate. Recently, heuristic optimization techniques have exhibited remarkable performance in a wide variety of problems from diverse fields due to the following advantages: simplicity, flexibility, derivation-free mechanism and local optima avoidance. For this reason, it has expanded tremendously. Inspired by different nature phenomena, scholars have proposed many successful and effective optimization methods. According to inspiration, these existing paradigms can be classified into three main categories: evolution-based, physics-based and swarm-based methods.

Evolution-based methodologies are inspired from the laws of biological evolution. The most popular evolution approach in this category is Genetic Algorithm (GA) [2], which imitates the theory of Darwinian evolution. Biogeography-Based Optimizer (BBO) on natural biogeography [3] and Bird Mating Optimizer (BMO) on natural evolution [4].

Physics-based algorithms are those who mimic the physical regulations of the universe, such as Simulated Annealing (SA) on the metallurgic annealing process [5], Ray Optimization (RO) on the Snell's light refraction law [6], Gravitational Search Algorithm (GSA) on the law of gravity and mass interactions [7], and Black hole (BH) on black hole phenomenon [8].

Swarm-based techniques simulate all kind of animal or human

* Corresponding author.
E-mail addresses: hgd_qy@126.com (Q. Zhang), wangrgui@hfut.edu.cn (R. Wang), yangjuan6985@163.com (J. Yang), dingkai0422@126.com (K. Ding), lyf_strawberry@sina.com (Y. Li), hgdhujgen@163.com (J. Hu).

http://dx.doi.org/10.1016/j.neucom.2016.09.068
Received 25 December 2015; Received in revised form 25 August 2016; Accepted 27 September 2016
Available online 03 October 2016
0925-2312/

behaviors. For instance, Particle Swarm Optimization (PSO) [9] on the foraging behavior of bird flocking, Flower Pollination algorithm (FPA) on the pollination process of flowers [10], Cuckoo Search (CS) on the brood parasitism of cuckoo species [11], Crow Search Algorithm (CSA) on the behavior of crows [12], Interior Search Algorithm (ISA) on interior design and decoration [13], Artificial bee colony (ABC) on the foraging behavior of honey bee swarm [14], and Harmony Search (HS) on the principle of music improvisation [15].

Regardless of a great number of new algorithms and their applications in different fields, there is a question here that if more techniques are needed in this field. The answer is positive according to the No Free Lunch (NFL) theorem [16], which logically proves that there is no optimization methodology for solving all optimization tasks. This theorem encourages the proposal of new methods with the hope to solve a wider range of problems or some specific types of unsolved problems. In addition, it is generally known that the population-based algorithm updates the candidate solutions based on their fitness values (information of the search space). Undoubtedly, accurate spatial information are great beneficial to find a rough approximation of the global optimum. However, a great majority of population-based methods tend to share a common characteristic, each agent intends to generate only one new individual in each iteration. So, the search space can't be systematically exploited or explored, owing to a paucity of new solutions robust enough to guide search. This motivates us to study whether a new heuristic algorithm can be designed by combining a new vector generation strategy and improving the amount of the candidate solutions with precious space messages. In the other words, whether the search space around each member can be sampled systematically through increasing the number of selectable solutions. Motivated by the mentioned perspectives, authors attempt to get inspiration from the decision-making behavior of human and design a new optimization methodology.

This paper along the motivation proposes a newly heuristic paradigm, collective decision optimization algorithm (CDOA), which is inspired from the decision-making behavior of human. Like other nature-based methods, CDOA is also a population-based search technique that uses a population of candidate solutions to proceed to the global optimum. At each iteration, guided by different locations, an agent yields several promising solutions on multi-step position-selected scheme [17], which refers to the new location generated by the previous movement will be defined as the start point for the next movement guided by the other position. For the global best, it is beneficial to slightly change its position by the random walk, which works as a local search. In such a case, several candidate solutions are randomly generated and placed around the global optimal solution. As a result, increasing the number of comparable solutions for each agent benefits to sample the search space systematically and search the global optimal solution preferably.

The rest of this paper is summarized as follows. Section 2 reviews the inspiration and principles of the proposed algorithm. In Section 3, numerical experiment is carried out on a comprehensive set of benchmark functions. Section 4 investigates the effectiveness of the proposed CDOA methodology in training artificial neural networks. The main work of this study is concluded in Section 5.

2. Collective decision optimization algorithm (CDOA)

In this section, to describe the proposed methodology intuitively, the inspiration of CDOA is first reviewed briefly. Then, the basic calculation model and pseudo code are then outlined in detail.

2.1. Inspiration

The decision-making behavior of human is the main inspiration of the proposed technique. When faced with a questionable problem or doubt, this action usually occurs to unite group members with different abilities adequately to develop a best program or plan. In our life, a typical decision action is holding a meeting, which is selected as example to explain the inspiration. As shown in Fig. 1, each member of the meeting is called decider, who corresponds to a thought or plan. In the discussion process of meeting, everyone will express and exchange their own thoughts or plans. The best one among the resultant schemes is selected as the final result.

In fact, developing a good plan is not a simple process, since this process usually involves some influence factors. For instance, some researchers firmly believe that group thinking is a significant concept in the study of collective decision-making behavior, penetrates the entire process of group decision-making, and affects the decision behavior to some extent [18–20]. Conformity behavior is one of the typical forms of collective thinking. Other influence factors include experience, leader, other decider's viewpoint and innovation [21–31]. These components also play importance role in the decision-making process. It is worth mentioning that these factors can be combined arbitrarily. Any one or more of the five factors can generate good result. Hence, multi-step position-selectable search scheme is employed to design operators. The mathematical model is explained in detail in next section. Table 1 presents the corresponding relationships between the inspiration and optimization process.

2.2. Mathematical model and algorithm

2.2.1. Group generation

In order to simulate the decision-making behavior clearly, we suppose an initial population with N agents is randomly sampled from the feasible solution space.

$$X_i(t) = (x_i^1(t), x_i^2(t), \ldots, x_i^D(t)), \; i = 1, 2, \ldots, N \tag{1}$$

Fig. 1. The phenomenon of making plans to problems in the meeting.

Table 1
The corresponding relations between the inspiration and optimization process.

Optimization process	Inspiration
Population	Meeting
Population size	The number of decider
Agents	Deciders or Plans
Feasible solutions	Plans
Fitness value	The quality of plan
Global optimal solution	The best plan

$$x_i^k(t) = LB^k + r \times (UB^k - LB^k),\ k = 1, 2, \ldots, D \quad (2)$$

$$Pop(t) = (X_1(t), X_2(t), \ldots, X_N(t))$$

where N is the population size, D denotes the dimension of optimization problem, t is the current number of generation, r is a random number between 0 and 1, LB and UB are the lower and upper bounds of variables, respectively.

2.2.2. Experience-based phase

In the meeting, for a issue, decider's first reaction is to ponder and develop a preliminary plan based on personal experience accumulated from daily life. In CDOA, personal experience is defined as the best position of individual (φ_P) so far. The operator is expressed as follows:

$$newX_{i_0} = X_i(t) + \overrightarrow{\tau_0} \times step_{size}(t) \times d_0 d_0 = \varphi_P - X_i(t) \quad (3)$$

where $\overrightarrow{\tau_0}$ is a random vector with each member in the range (0, 1), $step_{size}(t)$ denotes the step size of the current iteration, d_0 is the direction of movement.

2.2.3. Others'-based phase

After the experience-based phase, deciders already have their own thoughts or plans. They will interact randomly with other members in the meeting. we know a decider can receive something new if the other decider has better thought than him or her with the help of discussions and communications. In CDOA model, an individual ($X_j(t)$) is randomly selected from the population, and it is better than the current member ($X_i(t)$) in terms of fitness value. The calculation formula is designed as follows:

$$newX_{i_1} = newX_{i_0} + \overrightarrow{\tau_1} \times step_{size}(t) \times d_1 d_1 = beta_1 \times d_0 + beta_{11} \times (X_j(t) - X_i(t)) \quad (4)$$

where j denotes a random integer in the range [1,N]. $\overrightarrow{\tau_1}$ is a random vector with each number uniformly distributed in the interval (0, 1), $step_{size}(t)$ denotes the step size of the current iteration, d_1 is the new direction of movement, $beta_1$ and $beta_{11}$ are the random numbers in the range (−1, 1) and (0, 2), respectively.

2.2.4. Group thinking-based phase

In the meeting, everyone can express their own standpoints optionally, then, the decision of each decider will be influenced by the trend of the collective thinking. In proposed model, for the sake of simplicity, it may be supposed that the geometric center (φ_G) of all individuals is defined as the position of group thinking. From the statistical standpoint, the geometric center is a important digital characteristic and represents the changing trend of population on some level.

$$\varphi_G = 7\frac{1}{N}(X_1(t), X_2(t), \ldots, X_N(t)) = \left\{\frac{1}{N}\sum_{i=1}^{N} x_i^1(t), \frac{1}{N}\sum_{i=1}^{N} x_i^2(t), \ldots, \frac{1}{N}\sum_{i=1}^{N} x_i^D(t)\right\} \quad (5)$$

Then, the new position of agent is calculated using the following formula:

$$newX_{i_2} = newX_{i_1} + \overrightarrow{\tau_2} \times step_{size}(t) \times d_2 d_2 = beta_2 \times d_1 + beta_{22} \times (\varphi_G - X_i(t)) \quad (6)$$

where $\overrightarrow{\tau_2}$ is a random vector with each number uniformly generated in the interval (0, 1), $step_{size}(t)$ denotes the step size of the current iteration, d_2 is the new direction of movement, $beta_2$ and $beta_{22}$ are the random numbers in the range (−1, 1) and (0, 2), respectively.

2.2.5. Leader-based phase

As one of the primary deciders, leader plays an important role in the

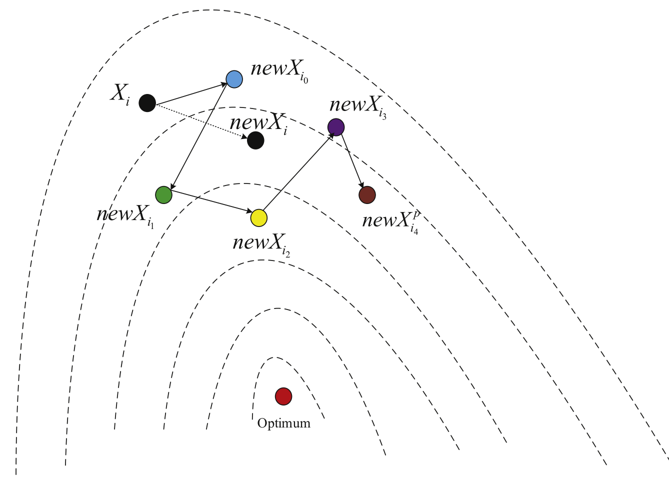


Fig. 2. Single individual searching trajectory.

Table 2
The comparisons between CDOA and PSO.

Characteristic	CDOA	PSO
Inspiration	The collective decision behavior of human	Animal swarm behavior
Search strategy	multi-step position-selectional search mechanism	Velocity updating Position updating Inspiration
Suitable task	Continuous problems	Continuous problems
Categorization	Population-based method	Population-based method
Guidance way	Guided by the notion of fitness	Guided by the notion of fitness

Table 3
Benchmark functions in experiments.

Test function	D	S	Minimum
f_1: Sphere Function	30	$[-100, 100]^D$	0
f_2: Schwefel's Problem 2.21	30	$[-100, 100]^D$	0
f_3: Generalized Rosenbrock's Function	30	$[-30, 30]^D$	0
f_4: Quartic Function i.e. Noise	30	$[-1.28, 1.28]^D$	0
f_5: Generalized Rastrigin's Function	30	$[-5.12, 5.12]^D$	0
f_6: Ackley's Function	30	$[-32, 32]^D$	0
f_7: Generalized Penalized Function	30	$[-50, 50]^D$	0
f_8: Shifted Sphere Function	30	$[-100, 100]^D$	0
f_9: Shifted Schwefel's Problem 1.2	30	$[-100, 100]^D$	0
f_{10}: Shifted Rotated High Conditioned Elliptic Function	30	$[-100, 100]^D$	0
f_{11}: Shifted Schwefel's Problem 1.2 with Noise in Fitness	30	$[-100, 100]^D$	0
f_{12}: Schwefel's Problem 2.6 with Global Optimum on Bounds	30	$[-100, 100]^D$	0
f_{13}: Shifted Rosenbrock's Function	30	$[-100, 100]^D$	0
f_{14}: Shifted Rotated Griewank's Function without Bounds	30	$[-600, 600]^D$	0
f_{15}: Shifted Rotated Ackley's Function with Global Optimum on Bounds	30	$[-32, 32]^D$	0
f_{16}: Shifted Rastrigin's Function	30	$[-5, 5]^D$	0
f_{17}: Shifted Rotated Rastrigin's Function	30	$[-5, 5]^D$	0
f_{18}: Shifted Rotated Weierstrass Function	30	$[-0.5, 0.5]^D$	0
f_{19}: Schwefel's Problem 2.13	30	$[-\pi, \pi]^D$	0
f_{20}: Shifted Expanded Griewank's plus Rosenbrock's Function (F8F2)	30	$[-3, 1]^D$	0
f_{21}: Shifted Rotated Expanded Scaffer's F6 Function (F8F2)	30	$[-100, 100]^D$	0

Table 4
Experiment results of all methods on benchmark functions.

Function	Index	CDOA	CS	CSA	FPA	PSO	ABC	GSA
f_1	mean	4.6268e−206	3.4920e−012‡	1.6924e−012‡	1.4890e−004‡	4.3654e−184‡	5.6754e−060‡	9.3674e−018‡
	std	0	1.7923e−012	1.4453e−012	2.0376e−004	0	6.4734e−030	2.2223e−018
	p	–	1.4157e−009	1.4157e−009	1.4157e−009	7.3796e−009	1.4157e−009	1.4157e−009
	h	–	1	1	1	1	1	1
f_2	mean	2.7420e−014	1.2507e−001‡	7.7452e−002‡	1.3363e+001‡	8.2481e−008‡	2.9946e+001‡	1.7646e−009‡
	std	4.8941e−014	6.9641e−002	6.5952e−002	2.6499e+000	9.4070e−008	5.1729e+000	2.5380e−010
	p	–	1.4157e−009	1.4157e−009	1.4157e−009	1.4157e−009	1.4157e−009	1.4157e−009
	h	–	1	1	1	1	1	1
f_3	mean	9.2071e−005	2.3123e+001‡	5.9593e+001‡	6.4993e+002‡	2.7183e+001‡	5.3045e−001‡	1.9214e+001‡
	std	1.9071e−004	1.5265e+000	2.1720e+001	1.6376e+003	5.5603e−001	7.8003e−001	2.1732e−001
	p	–	1.4157e−009	1.4157e−009	1.4157e−009	1.4157e−009	1.4157e−009	1.4157e−009
	h	–	1	1	1	1	1	1
f_4	mean	7.4010e−004	9.6952e−003‡	6.7767e−003‡	9.8189e−002‡	2.9475e−003‡	2.1239e−001‡	1.2372e−002‡
	std	2.9966e−004	2.5921e−003	2.5778e−003	6.2671e−002	1.6700e−003	4.3485e−002	2.6095e−003
	p	–	1.4157e−009	1.4157e−009	1.4157e−009	1.8356e−008	1.4157e−009	1.4157e−009
	h	–	1	1	1	1	1	1
f_5	mean	3.5839e−002	5.1585e+001‡	2.3202e+001‡	2.3929e+001‡	5.0092e+001‡	1.4211e−016†	1.3611e+001‡
	std	1.7919e−001	8.6703e+000	1.8453e+001	6.6789e+000	2.4420e+001	4.9158e−016	3.3213e+000
	p	–	1.3555e−009	1.3555e−009	1.3555e−009	1.3555e−009	1.3555e−009	1.3543e−009
	h	–	1	1	1	1	1	1
f_6	mean	8.1002e−015	6.5732e−004‡	3.4182e+000‡	4.8040e+000‡	1.7764e−014‡	4.4906e−014‡	2.0392e−009‡
	std	3.4809e−015	1.4000e−003	9.2831e−001	1.2929e+000	5.0243e−015	4.4563e−015	3.2676e−010
	p	–	3.1559e−010	3.1559e−010	3.1559e−010	9.1158e−009	3.1559e−010	3.1559e−010
	h	–	1	1	1	1	1	1
f_7	mean	1.3695e−032	7.4506e−011‡	1.9450e−002‡	1.5749e+001‡	4.3949e−003‡	1.3498e−032≀	9.4234e−019‡
	std	9.8608e−034	6.1194e−011	3.0203e−002	1.1116e+001	9.5152e−003	5.5867e−048	2.9110e−019
	p	–	1.3762e−010	1.3762e−010	1.3762e−010	5.2081e−006	3.3705e−001	1.3762e−010
	h	–	1	1	1	1	0	1
f_8	mean	6.0080e−029	7.6664e−012‡	6.0865e+002‡	2.5471e−003‡	5.5931e−027‡	0†	8.6771e−018‡
	std	1.6864e−028	5.3369e−012	7.3227e+002	3.4612e−003	1.0210e−026	0	2.2554e−018
	p	–	7.4638e−010	7.4638e−010	7.4638e−010	1.9947e−009	1.1742e−003	7.4638e−010
	h	–	1	1	1	1	1	1
f_9	mean	3.1507e−024	8.1626e+000‡	2.0157e+002‡	1.2079e+000‡	7.0147e−010‡	6.3692e+003‡	2.4338e+002‡
	std	4.6887e−024	2.4337e+000	3.4603e+002	1.1903e+000	1.4374e−009	1.3573e+003	6.0846e+001
	p	–	1.4157e−009	1.4157e−009	1.4157e−009	1.4157e−009	1.4157e−009	1.4157e−009
	h	–	1	1	1	1	1	1
f_{10}	mean	2.3376e+005	3.9865e+006‡	2.5248e+006‡	6.8036e+004†	1.2340e+006‡	9.8423e+006‡	2.2094e+006‡
	std	1.1911e+005	9.8687e+005	3.6624e+006	4.6408e+004	8.0957e+005	2.5382e+006	1.6153e+006
	p	–	1.4157e−009	3.6690e−009	9.5133e−008	1.8356e−008	1.4157e−009	1.4157e−009
	h	–	1	1	1	1	1	1
f_{11}	mean	4.2912e+000	4.6164e+003‡	2.4915e+003‡	1.6722e+004‡	2.5028e+002‡	5.4198e+004‡	2.2462e+004‡
	std	1.1908e+001	1.7171e+003	1.5329e+003	6.4535e+003	4.2950e+002	7.2137e+003	3.6615e+003
	p	–	1.4157e−009	1.4157e−009	1.4157e−009	3.9965e−008	1.4157e−009	1.4157e−009
	h	–	1	1	1	1	1	1
f_{12}	mean	2.6584e+003	3.3496e+003‡	8.8630e+003‡	1.4571e+003†	3.0612e+003≀	1.1428e+004‡	5.8377e+003‡
	std	5.4054e+002	6.1660e+002	5.6139e+003	6.7908e+002	1.0261e+003	1.8085e+003	1.3207e+003
	p	–	4.4489e−004	2.5677e−008	2.7218e−007	1.3017e−001	1.4157e−009	1.4157e−009
	h	–	1	1	1	0	1	1
f_{13}	mean	3.0420e−001	3.7427e+001‡	6.5769e+007‡	3.5209e+003‡	3.7232e+001‡	2.5299e+000‡	3.4410e+002‡
	std	8.9934e−001	2.7330e+001	2.2094e+008	6.8948e+003	4.0984e+001	2.8435e+000	3.3954e+002
	p	–	1.4157e−009	1.4157e−009	1.4157e−009	5.8547e−009	4.5414e−007	1.4157e−009
	h	–	1	1	1	1	1	1
f_{14}	mean	1.6338e−002	7.4986e−003†	1.1283e+002‡	3.1614e−001‡	3.4386e−002‡	4.1772e−002‡	4.5295e+002‡
	std	1.1822e−002	1.0493e−002	1.2616e+002	3.2347e−001	2.3443e−002	9.3026e−003	2.5041e+002
	p	–	3.0726e−004	1.4131e−009	1.5938e−009	6.3767e−004	1.7929e−007	1.4131e−009
	h	–	1	1	1	1	1	1
f_{15}	mean	2.0892e+001	2.0931e+001‡	2.0935e+001‡	2.0901e+001≀	2.0974e+001‡	2.0881e+001≀	2.0040e+001†
	std	3.3071e−002	6.6805e−002	7.0369e−002	6.0863e−002	5.7450e−002	5.0586e−002	2.8416e−002
	p	–	6.8504e−004	8.3203e−003	5.0945e−001	6.3805e−004	4.7281e−001	1.4157e−009
	h	–	1	1	0	1	0	1

(continued on next page)

Table 4 *(continued)*

Function	Index	CDOA	CS	CSA	FPA	PSO	ABC	GSA
f_{16}	mean	6.0299e−001	6.1656e+001‡	1.3775e+002‡	7.9600e+001‡	8.6920e+001‡	7.1054e−017†	3.0326e+001‡
	std	2.2056e+000	9.6049e+000	2.2813e+001	2.9603e+001	1.7537e+001	3.5527e−016	6.6255e+000
	p	–	1.4157e−009	1.4157e−009	1.4157e−009	1.4157e−009	1.3762e−010	1.4157e−009
	h	–	1	1	1	1	1	1
f_{17}	mean	4.0996e+001	1.4143e+002‡	2.0028e+002‡	2.1722e+002‡	1.2438e+002‡	3.6417e+002†	2.1292e+001†
	std	1.2470e+001	2.4067e+001	4.9769e+001	5.2955e+001	3.8902e+001	4.5320e+001	5.5322e+000
	p	–	1.4157e−009	1.4157e−009	1.4157e−009	1.5967e−009	1.4157e−009	1.3026e−008
	h	–	1	1	1	1	1	1
f_{18}	mean	3.2481e+001	2.7825e+001‡	3.2385e+001≀	2.6733e+001†	3.1582e+001≀	3.1030e+001†	6.3695e−002†
	std	2.8385e+000	2.0823e+000	3.7362e+000	2.1781e+000	4.5819e+000	2.0542e+000	3.1526e−001
	p	–	3.5302e−006	8.6137e−001	2.7218e−007	6.6948e−001	1.1657e−002	1.4157e−009
	h	–	1	0	1	0	1	1
f_{19}	mean	1.6085e+003	3.2576e+004‡	2.3862e+004‡	8.7906e+003‡	7.1874e+003‡	1.1399e+004‡	2.9459e+003‡
	std	2.2122e+003	9.5535e+003	2.1123e+004	1.0381e+004	6.7243e+003	4.1582e+003	3.5641e+003
	p	–	1.4157e−009	2.2967e−008	5.6183e−006	1.2675e−005	7.3796e−009	1.0432e−002
	h	–	1	1	1	1	1	1
f_{20}	mean	1.5895e+000	7.5051e+000‡	6.7026e+000‡	6.3731e+000‡	4.1861e+000‡	2.0390e+000‡	4.6508e+000‡
	std	6.0932e−001	1.0032e+000	1.9560e+000	2.0220e+000	1.3834e+000	1.9697e−001	8.2339e−001
	p	–	1.4157e−009	1.8002e−009	1.5967e−009	1.0414e−008	3.5681e−004	1.4157e−009
	h	–	1	1	1	1	1	1
f_{21}	mean	1.1254e+001	1.2932e+001‡	1.2751e+001‡	1.2500e+001‡	1.2802e+001‡	1.3291e+001‡	1.4044e+001‡
	std	5.8315e−001	2.3189e−001	4.1826e−001	3.9114e−001	3.0557e−001	1.5605e−001	2.3216e−001
	p	–	1.4157e−009	8.2805e−009	3.2057e−008	2.8908e−009	1.4157e−009	1.4157e−009
	h	–	1	1	1	1	1	1
	‡	–	20	20	17	19	14	18
	†	–	1	0	3	0	5	3
	≀	–	0	1	1	2	2	0

‡, † and ≀ indicate that the performance of CDOA is better than, worse than and similar to that of the corresponding algorithm, respectively.

overall decision procedure. It not only brings different influences to other policymakers, but also determines the direction and final result of decision. In the proposed model, Leader (φ_L) is regarded as the best individual (the fittest element) in population.

$$newX_{i_3} = newX_{i_2} + \overrightarrow{\tau_3} \times step_{size}(t) \times d_3 d_3 = beta_3 \times d_2 + beta_{33} \times (\varphi_L - X_i(t)) \tag{7}$$

where $\overrightarrow{\tau_3}$ is a random vector with each number uniformly generated in the interval (0, 1), $step_{size}(t)$ denotes the step size of the current iteration, d_3 is the new direction of movement, $beta_3$ and $beta_{33}$ are the random numbers in the range (−1, 1) and (0, 2), respectively.

Besides that, for convenience, we assume that the leader's thought or program can only be changed arbitrary by himself. In our model, it is advantageous to slightly change its position using the random walk strategy, which works as a local search. In this case, some neighbors can be generated randomly around the best solution.

$$newX_q = \varphi_L + \overrightarrow{W_q} \quad (q = 1, 2, 3, 4, 5) \tag{8}$$

where $\overrightarrow{W_q}$ is a random vector with each number in the range (0, 1).

2.2.6. Innovation-based phase

It is generally known that innovation not only breaks the bonds of convention, but also broadens our horizons. Some scholars believe that it is also another effective pathway to generate good scheme in the decision process. In our model, innovation refers to make a small change among variables. This is equivalent to a one-dimensional mutation operator in evolutionary methods. This operator can be implemented as follows:

$$r_1 \le MF newX_{i_4} = newX_{i_3} newX_{i_4}^p = LB(p) + r_2 \times (UB(p) - LB(p)) \tag{9}$$

where r_1 and r_2 are two random values distributed in the interval (0, 1), p is randomly generated in the range [1, D], MF is the innovation (mutation) factor, which is defined as a large value for improving the population diversity to prevent premature convergence in this study.

Another concern here is that the updating of step size with respect to iterations (t). Large search step size ($step_{size}(t)$) can make the generated vectors distribute widely in the search space and can explore the search space effectively in the initial stage of evolution. In the rest of iterations, the small step size makes the search focus on neighborhoods of the solutions, and thus it can speed up the convergence. Therefore, this study employ an adaptive mechanism described blow.

$$step_{size}(t) = 2 - 1.7\left(\frac{t-1}{T-1}\right) \tag{10}$$

where T is the maximum number of iteration.

As described above, the main pseudo code of the CDOA algorithm is summarized as follows:

Algorithm 1. Collective Decision Optimization Algorithm

Initialize a population (*Pop*) and termination criterion (T)
Calculate the fitness of each search agent
The personal best location ←*Pop*, t=1;
while the termination criterion is not satisfied ($t \le T$) **do**
  Find the global best (φ_L)
  Compute the step size ($step_{size}(t)$) on Eq.(10)
for $i = 1 \to N$ **do**
    $newpop \leftarrow []$
    if φ_L **then**
      Calculate the new solutions ($newX_q$) by the Eq. (8)
      $newpop \leftarrow newX_q$
    else
      Change the location of an agent by Eq. (3)-(7), (9),
      $newpop \leftarrow [newX_{i_0}, newX_{i_1}, newX_{i_2}, newX_{i_3}, newX_{i_4}^p]$

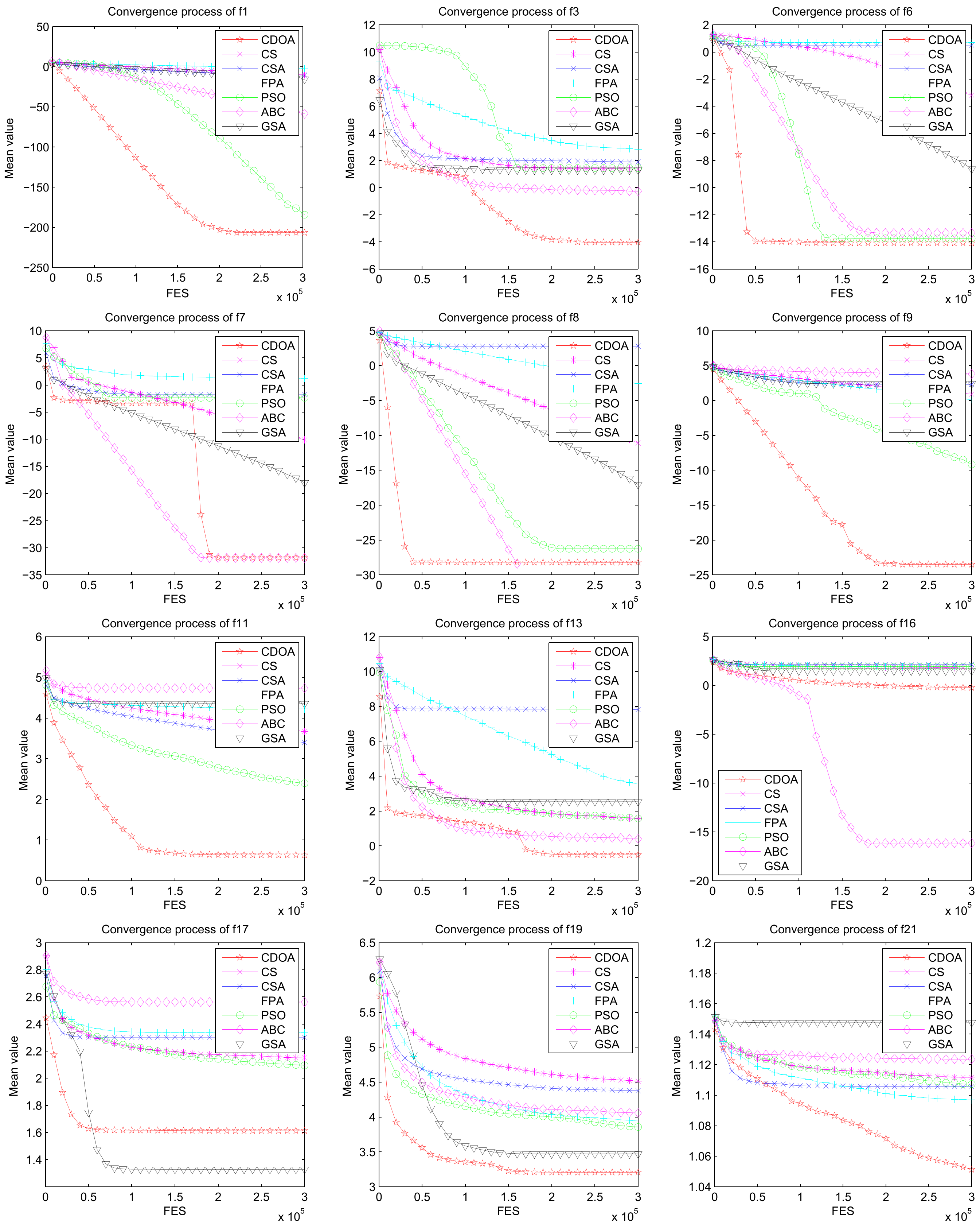


Fig. 3. The convergence process of all methods for some test functions.

end if
Evaluate the fitness of *newpop*
Update $X_i(t)$ and the personal best location using the best one among these resultant positions
end for
t=t+1;
end while
Output the best candidate solution

2.2.7. Discussion

Like other population-based methods, CDOA also uses a population of candidate solutions generated randomly in the search space to proceed to the global optimum. At each generation, guided by individual best location, the other member's location, the geometric center of all members, global best position and mutation operator, each agent can move to several promising regions on multi-step position-selectional technique, which means that the next movement starts with the location generated by the previous movement. Besides that, the random walk, a local search strategy, is employed to yield some promising points around the best individual. That is, each agent will generate several potential solutions at each iteration. The best one among these possible promising solutions (including the current agent) will be used for the next search.

Inspecting the proposed algorithm carefully, the whole evolution process involves three essential components: selection of the best, randomization via multi-step position-selectable strategy and local random walk. Keeping the best candidate solutions equates with some forms of elitism commonly employed in EAs, which maintains the best agent can be retained to the next iteration. In addition, the discovery of new candidate solutions not only keeps the diversity of population, but also replaces some of worse agents to proceed to the optimum.

However, to prevent premature convergence during the process of evolution, it is necessary to sample the whole search space systematically and generate new solutions diverse enough. This study adopts multi-step position-selectional search mechanism to guarantee that attracted by different positions, each agent can move to other places randomly, and generate some potential solutions distributed in the search space. This not only increases the number of new solutions, broadens the search trajectory, but also improves the probability of finding the optimal solution. Requiring the search agent to move toward different positions benefits higher exploration of the search space and lower probability of local optima stagnation. Another way of improving the diversity slightly is mutation operator (Eq. (9)) involved innovation-based phase. It causes sudden change of the agents in search space and promotes exploration. These mentioned contents can guarantee the search members to escape from local optima easily.

In fact, the employed design strategy can also ensure the search agents converge to the best one for exploiting the search space. Besides that, the exploitation around the best solution is also implemented by random walk, in which the exploitation moves within the neighborhood of the best solutions locally. Another way of intensifying local search slightly is to apply the current best to replace the previous location. In this way, the local exploitation strategy is able to find some promising areas around the best one found so far effectively. Note that the step size ($step_{size}(t)$), a adaptive strategy, is designed to adjust the exploration and exploitation of the algorithm preferably. Therefore, the above-mentioned contents can keep a good tradeoff between exploration and exploitation.

The following highlights the similarities and differences between CDOA and PSO. On the one hand, they draw inspiration from completely different phenomena. PSO is inspired by animal swarm behavior, fish schooling and bird flocking. While the inspiration of CDOA originates from the decision behavior of human. On the other hand, the updating formulations and generation of new candidates in CDOA are completely different from those used in PSO. The updating formulations of each particle in PSO involves a velocity item. Nevertheless, the concept does not appear in CDOA. As shown in Fig. 2, in PSO, it can be assumed that the current agent (X_i) generates only one new individual ($newX_i$) on the evolution mechanism. The selection operator is performed to select the better one (X_i or $newX_i$) to enter the next generation. For each particle, the whole updating process only involves two points in each iteration. While the proposed model adopts multi-step position-selectional search mechanism, including experience-based operator, others'-based operator, group thinking-based operator, leader-based operator and innovation-based operator, which guides the agent (X_i) to move to some possibly promising regions ($newX_{i_0}$, $newX_{i_1}$, $newX_{i_2}$, $newX_{i_3}$, $newX_{i_4}^p$). The best one (X_i or anyone of the possible positions) among these resultant positions will be passed onto the next generation. It is obvious that there exists fundamental difference between CDOA and PSO.

Although there are some differences between them, CDOA still shares some similarities with PSO. Firstly, similar to PSO, the proposed method aims to solve continuous optimization tasks effectively. Secondly, both of them utilize the notion of fitness to search better candidate solutions. Finally, like most of swarm intelligence algorithms, CDOA is also a population-based and nature-based method. The comparisons between them are further summarized in Table 2.

3. Numerical experiments

Although these observations guarantee that the proposed method is able to improve the initial random solutions and convergence to a better point in the search space theoretically, it is necessary to verify the performance of CDOA in practice.

3.1. Benchmark functions

It is a common in this field to investigate the effectiveness of algorithms using a set of mathematical functions with known global optima. In this study, 21 benchmark functions with 30 variables listed in Table 3 are employed as test beds for comparison. The detailed description of these problems can be found in the literature [32,33].

3.2. Experimental platform and performance metric

All experiments are conducted on the same platform with Inter Xeon E3-1226 3.30 GHz, 8 GB memory, Windows 10 operating system and MatlabR2009a.

In experimental studies, the mean error value (f_{mean}) and standard deviation (*std*) of the best solution obtained in the last

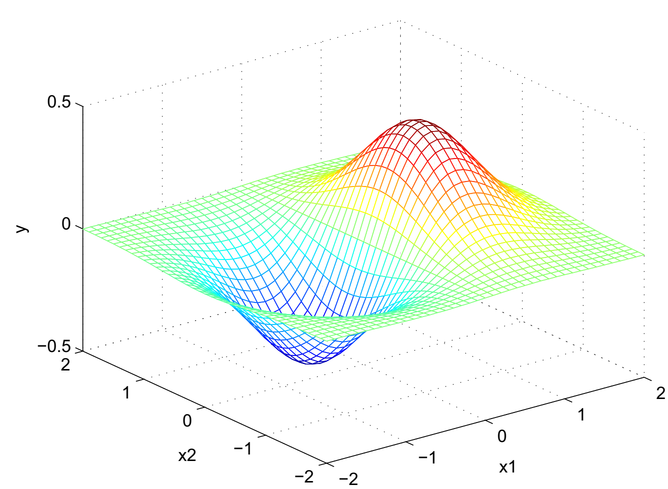


Fig. 4. 3-D contours of the Peak function.

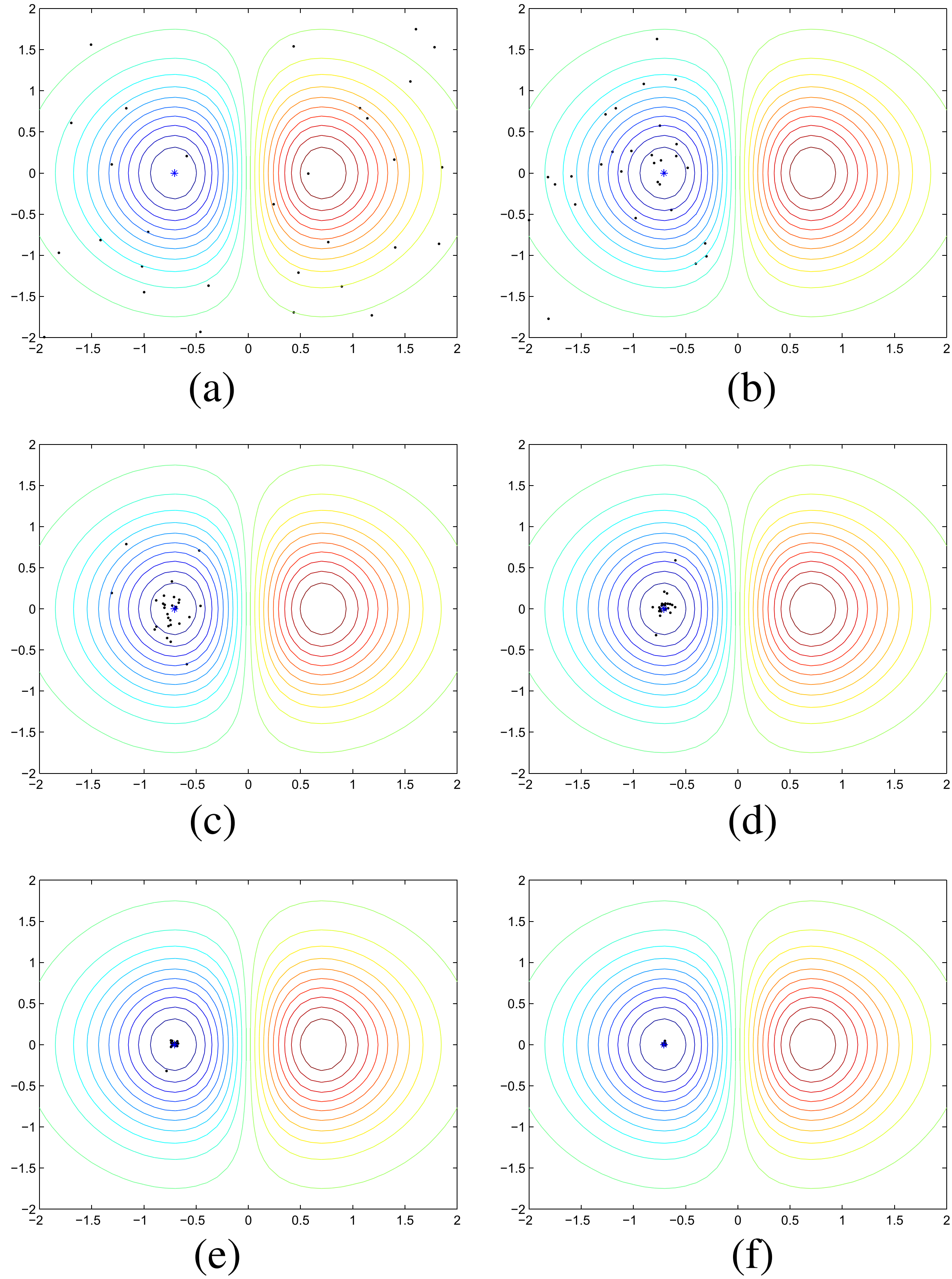


Fig. 5. The positions of the search agents after (a) 1st iteration, (b) 3rd iteration, (c) 5th iteration, (d) 7th iteration, (e) 9th iteration, (f) 11th iteration.

iteration are conducted as the performance metrics to evaluate the search capability of each approach. In addition, in order to determine whether the results generated by CDOA are statistically different from the results obtained by other algorithms, the Wilcoxon's rank sum tests [34] at a 0.05 significance level are conducted between the CDOA's result and the best result achieved by other methodologies for each problem.

3.3. Parameter settings

In order to ensure computation and comparison as fair as possible, the maximum number of function evaluations (*MaxFES*) is regarded as the termination condition, which is set to D×10,000 for each method. The mean error values at specified checkpoint $(0.1, 1, 2, \ldots, 29, 30) \times 10,000$ will be recorded. Moreover, to evaluate the performance of the proposed methodology roundly, some of the well-known and recent algorithms are chosen: PSO, CS, FPA, GSA, ABC and CSA. The following provides the parameter settings of CDOA and these compared methods.

There are three control parameters in the proposed algorithm, step size ($step_{size}(t)$) decays monotonically from 2 to 0.3 over the course iterations. The population size and mutation factor (*MF*) are set as 50 and 0.8, respectively. It should be noted that tuning these parameters is a trial-and-error procedure based on the experiment results. Since finding right parameter values in such a way is often very time-consuming. Therefore, the investigation about parameters needs further study in the future.

- PSO Settings [9]: population size=40, positive constant $c1 = c2 = 1.49445$, inertia weight w_{max}=0.9, w_{min}=0.4.
- CS Settings [11]: population size=50, discovery rate=0.25.
- FPA Settings [10]: population size=25, proximity probability=0.8.
- GSA Settings [7]: population size=50, α and G_0 are set to 20 and 100, respectively;
- ABC Settings [14]: population size=40, the number of food source-s=population size, number of onlookers=number of employed bees=0.5 × population size, limit=100.
- CSA Settings [12]: population size=50, *fl*=2, *AP*=0.1;

3.4. Experiment results

In this part, Table 4 summarizes the statistical results obtained in 25 independent runs by each method. The last three rows of Table 4 report the comparison results. Besides that, in order to compare the convergence rate of the proposed methodology with its competitors, Fig. 3 provides some convergence curves of the test functions.

Comparison with convergence precision: Unimodal functions ($f_1 - f_4$, $f_8 - f_{12}$) are fruitful for evaluating the exploitation capability of algorithm, since they have only one global optimum and no local optima. According to the results reported in Table 4, it is evident that none of the method can perfectly solve all test problems. The Wilcoxon's rank sum results, which are recorded in the last three rows of Table 4, provide information to statistically demonstrate that the proposed algorithm yields much better solutions than other methods. CDOA outperforms CS, CSA and GSA for all unimodal functions, and performs similar to PSO on f_{12}, slightly worse than ABC and FPA considering f_8, f_{10} and f_{12}, respectively. Whereas, from a statistical viewpoint, there is no distinct difference between the results obtained by CDOA and PSO for f_{12}. In addition, it should be noted that CDOA performs significantly better than its competitors on f_3, which is much difficult to be solved.

While multimodal functions ($f_5 - f_7$, $f_{13} - f_{21}$) involve a massive number of local optima whose number increases exponentially with the problem dimension. They are conducive to verify the exploration of algorithm, since this kind of test problems reflects a methods capability of escaping from poor local optima and locating a good near-global optimum. According to the Wilcoxons rank sum results of Table 4, it is apparent that CDOA performs significantly better than other algorithms for the majority of functions. In terms of the function f_7, the consistent superiority of the proposed algorithm over ABC is not observed here, while CDOA is highly superior to other techniques on the simulation results. The proposed algorithm is slightly worse than ABC and CS on function f_5, f_{16} and f_{14}, respectively. The performance of CDOA is superior to its peers, where CDOA performs equivalent to ABC and worse than GSA on f_{15}. For the function f_{17} where CDOA can generate better solutions than other optimization methods except ABC and GSA in the aspect of calculation accuracy. The proposed method performs exactly the same as CSA and PSO, slightly outperformed by FPA, ABC and GSA for the eighteen function.

Comparison with convergence rate: Many researchers usually intend to study the convergence behavior on the simple sphere model (f_1). As shown in Fig. 3, in the beginning, CDOA performs faster than other state-of-art optimization algorithms due to its excellent global search ability, which quickly guides agents to approximate the neighborhood of the global optimum, and reaches approximately e-200 level after about 2e+5FES, while PSO can only achieve approximately e-100 level. After that, the convergence speed becomes slow substantially due to its search is not sufficiently localized, while PSO guarantees a nearly constant convergence speed over the course iterations. The convergence behavior of CDOA on function f_1 also may be observed in f_6, f_8, f_9, f_{11} and f_{17}.

In addition, the largest difference in convergence characteristics between CDOA and its peers occurs with function f_3, the Rosenbrocks function, which is characterized by a narrow valley from the perceived local optima to the global optimum. Other algorithms perform poorly since they mainly search in a relatively small local neighborhood, which contains the points with the same fitness value. Therefore, it is not easy for them to move from the current state to a better one. Whereas, CDOA has wide search trajectory, which is apparent difference from other recent methodologies. Such advantage enables the proposed method to jump from one status to a better one with relative ease. The convergence characteristic of function f_3 also may be seen in f_7, f_{13} and f_{19}. In fact, inspecting the evolutionary processes of f_{16} and f_{21}, it is not difficult to find another phenomenon that other heuristic techniques are easy to fall into a poor local optimum quite early, while CDOA appears to converge at a good convergence rate and improves its solution steadily with the increasing number of iterations.

Search history: although these results indirectly indicate that the proposed technique is able to improve the initial solutions and converge to a point in the search space, we further calculate and discuss the search history to investigate the convergence of the proposed CDOA methodology. The experiment is implemented on a Peak function(see Fig. 4) [35], which is defined as:

$$y = x_1 e^{(-x_1^2 - x_2^2)},\ x_1, x_2 \in (-2, 2)$$

For this problem, 30 individuals are required to search the minima, the position in spread of the search agents by after 1st, 3rd, 5th, 7th, 9th and 11th iterations are respectively shown in Fig. 5 (a–f). These individuals are displayed by '·' marks and the global optimal is labeled by '*' mark. As it is seen, the sample points tend to explore promising

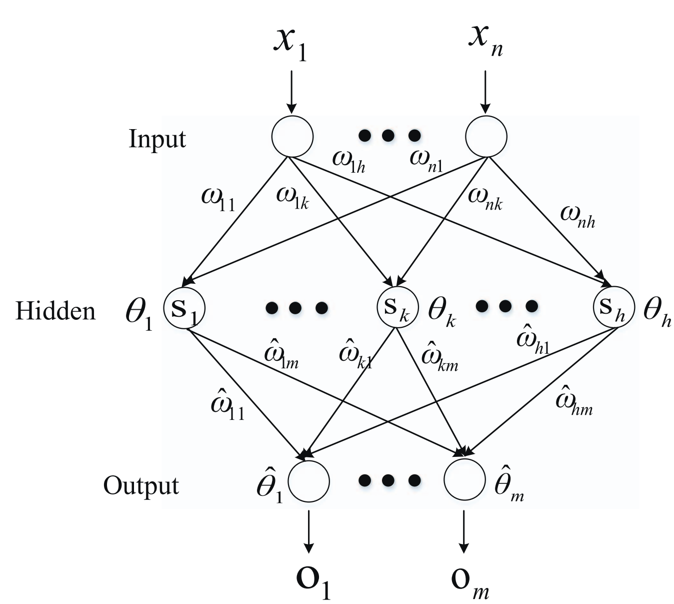


Fig. 6. The basic structure of FNNs.

regions of the search space and exploit very accurately around the global optima over the course iterations. The observations prove that the CDOA algorithm is effective in approximating the global optimum of optimization tasks.

As a summary, the simulation results of this section experimentally confirmed that the proposed method can generate competitive results and can perform significantly better than other well-known techniques. In addition, the convergence of CDOA is also experimentally analyzed on convergence curves and search history. It can be stated that the CDOA method is able to be effective in solving other problems as well.

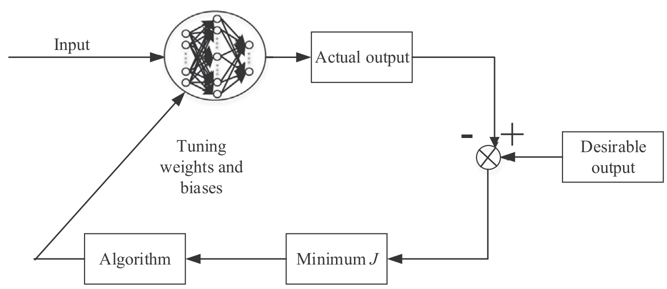


Fig. 7. The principle diagram of FNNs training.

Therefore, training artificial neural networks is required to further investigate the effectiveness of CDOA and provide a comprehensive study in the next section.

4. Problem statement

Feedforward neural networks (FNNs), the simplest structures of artificial network devised, consist of densely interconnected adaptive simple processing elements [36]. Recently, many practical applications of different fields, pattern recognition, function approximation and data compression, have been successfully solved by FNNs with three layers (including one input layer, one hidden layer, and one output layer). In addition, it has been proven that FNNs with three layers can approximate any continuous and discontinuous function [37]. Therefore, three layers FNNs are employed in this section to solve two nonlinear function approximation issues.

4.1. The principle of FNNs training

From the basic structure of FNNs with three layers (including one input layer, one hidden layer, and one output layer) depicted in Fig. 6, it can be verified that the process of information transmission from input layer to output layer can be implemented by the hidden layer

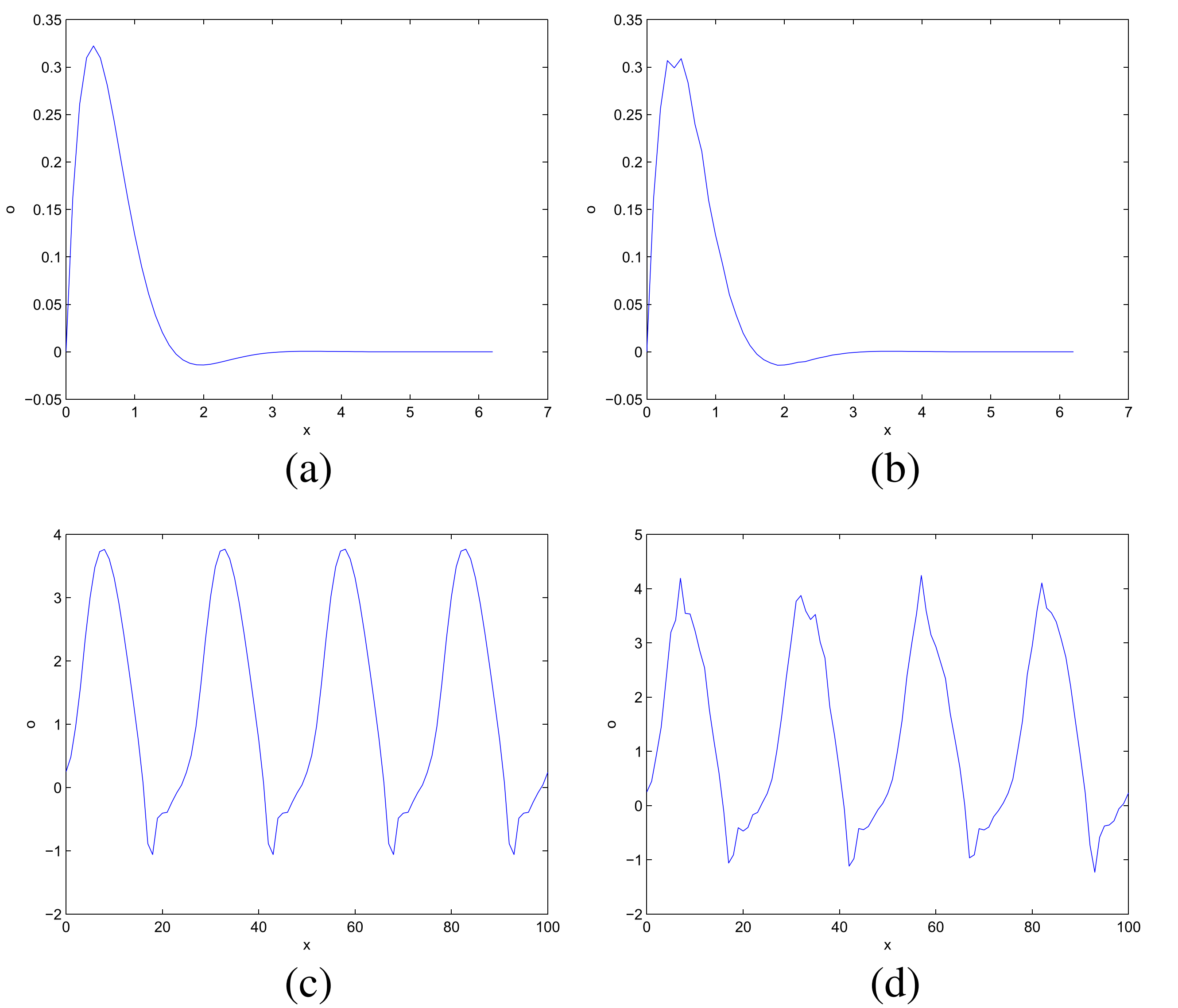


Fig. 8. The curves of two approximation problems (a) SISO without noise, (b) SISO with noise, (c) MISO without noise, (d) MISO with noise.

Table 5
Experiment results of all methods on different the number of the hidden layer nodes.

Function	Index	h=2	h=3	h=4	h=5	h=6	h=7	h=8	h=9	h=10
SISO without noise	mean	2.2986e−03‡	8.0861e−04≀	6.0299e−04	6.7059e−04≀	2.0731e−03‡	8.0592e−04≀	7.6750e−04≀	8.2023e−04≀	7.2008e−04≀
	std	2.2027e−03	9.5816e−04	4.7957e−04	5.2539e−04	1.8476e−03	6.0995e−04	1.0024e−03	8.9873e−04	7.0263e−04
	p	5.2508e−04	7.7102e−01	–	6.9797e−01	6.2435e−04	2.3658e−01	5.6051e−01	5.8694e−01	8.9196e−01
	h	1	0	–	0	1	0	0	0	0
	Testerror	3.4250e−04≀	3.2414e−03≀	1.9849e−03	1.2230e−03≀	4.6064e−03≀	1.3320e−03≀	1.9678e−03≀	2.8457e−03≀	4.4666e−03≀
	std	4.0241e−04	1.0250e−02	7.2891e−03	2.3564e−03	1.3713e−02	3.2083e−03	5.4835e−03	6.2906e−03	1.7969e−02
	p	8.9196e−01	7.2690e−01	–	2.9475e−01	9.3814e−01	7.7102e−01	9.0732e−01	3.1300e−01	5.0945e−01
	h	0	0	–	0	0	0	0	0	0
SISO with noise	mean	1.2931e−03‡	9.2798e−04‡	5.3076e−04	6.1936e−04≀	9.3804e−04‡	9.6281e−04‡	9.6567e−04‡	1.0411e−03‡	9.5201e−04‡
	std	1.1524e−03	7.3597e−04	6.3464e−04	5.1308e−04	7.1128e−04	9.7522e−04	7.7930e−04	9.9582e−04	7.4631e−04
	p	1.7925e−02	1.0006e−02	–	3.0378e−01	1.0432e−02	2.9651e−02	3.1262e−02	3.1262e−02	2.0766e−02
	h	1	1	–	0	1	1	1	1	1
	Testerror	9.7857e−03≀	3.8995e−03≀	5.8186e−03	1.9274e−03≀	5.4588e−04≀	6.2834e−04≀	1.1570e−03≀	1.4394e−02‡	1.3395e−03≀
	std	4.6174e−02	1.0534e−02	1.9988e−02	6.0317e−03	8.3976e−04	1.4624e−03	2.3916e−03	4.5250e−02	3.2600e−03
	p	6.8367e−01	6.9797e−01	–	8.4615e−01	8.4615e−01	7.8589e−01	8.4261e−01	4.4923e−02	9.2272e−01
	h	0	0	–	0	0	0	0	1	0
MISO without noise	mean	1.5083e−01≀	1.6357e−01≀	1.5956e−01	1.9601e−01≀	1.8693e−01≀	2.2601e−01‡	2.5137e−01‡	2.4287e−01‡	2.2040e−01‡
	std	6.4310e−02	8.6900e−02	6.1806e−02	9.2928e−02	9.0762e−02	1.1216e−01	1.3133e−01	1.5621e−01	1.2354e−01
	p	8.6137e−01	9.4213e−01	–	1.8064e−01	4.7281e−01	2.9771e−02	7.4153e−03	3.7239e−02	2.9771e−02
	h	0	0	–	0	0	1	1	1	1
	Testerror	1.8079e−01≀	1.9895e−01≀	2.1608e−01	2.6996e−01‡	2.1171e−01≀	2.8975e−01‡	2.7876e−01‡	2.7541e−01‡	2.7401e−01‡
	std	8.8379e−02	1.0762e−01	1.0450e−01	1.6539e−03	7.4694e−02	1.3638e−01	1.7126e−01	1.4749e−01	1.1711e−01
	p	2.9475e−01	5.7365e−01	–	4.1874e−02	7.1238e−01	3.7884e−02	4.8932e−02	4.8061e−02	4.7806e−02
	h	0	0	–	1	0	1	1	1	1
MISO with noise	mean	1.9003e−01≀	2.1398e−01≀	2.0572e−01	2.2772e−01≀	2.6338e−01‡	2.8021e−01‡	2.7446e−01‡	2.9551e−01‡	2.6905e−01‡
	std	5.0140e−02	5.2435e−02	6.3185e−02	7.2344e−02	9.5148e−02	1.3894e−01	9.2676e−02	1.1945e−01	1.0886e−01
	p	3.0378e−01	7.1238e−01	–	3.7211e−01	2.5659e−02	1.1160e−02	1.1029e−02	9.8635e−03	4.5663e−02
	h	0	0	–	0	1	1	1	1	1
	Testerror	2.3272e−01≀	2.8326e−01≀	2.8022e−01	3.2882e−01‡	3.4938e−01‡	3.3223e−01‡	3.5136e−01‡	3.9510e−01‡	3.9343e−01‡
	std	5.3254e−02	8.1574e−02	9.2702e−02	7.2723e−02	8.3424e−02	7.1919e−02	8.7960e−02	1.8335e−01	1.7818e−01
	p	7.4251e−02	8.3099e−01	–	5.4561e−03	6.4251e−03	7.3017e−03	5.8655e−03	8.8087e−03	2.2048e−02
	h	0	0	–	1	1	1	1	1	1
	‡	2	1	–	2	4	4	5	6	5
	†	0	0	–	0	0	0	0	0	0
	≀	6	7	–	6	4	4	3	2	3

‡, † and ≀ indicate that the performance of CDOA is better than, worse than and similar to that of the corresponding algorithm, respectively.

with connection weights and sigmoid activation function. The output of each hidden epoch can be calculated using the following calculation formula.

$$F(s_j) = \frac{1}{1 + exp(-s_j)} s_j = \sum_{i=1}^{n} \omega_{ij} x_i - \theta_j (j = 1, 2, \ldots, h) \tag{11}$$

where s_j is the *jth* hidden unit, n indicate the number of input units, ω_{ij} is the corresponding connection weight from the *ith* input unit to the *jth* hidden unit, θ_j is the bias of the *jth* hidden unit, and x_i and h denote the *ith* input variable and the number of hidden units, respectively

After that, the calculation results of output layer can be computed as follows:

$$o_p^i = \sum_{j=1}^{h} \hat{\omega}_{jp} F(s_j) - \hat{\theta}_p \tag{12}$$

where o_p^i, $(p = 1, 2, \ldots, m;\ i = 1, 2, \ldots, P)$ is the *pth* output result of *ith* input unit, m is the number of output units, P is the total number of training sampling points, $\hat{\omega}_{jp}$ is the corresponding connection weight from the *jth* hidden unit to *pth* output unit, and $\hat{\theta}_p$ is the bias of the *pth* output unit.

Without loss of generality, the learning error of FNNs training is defined as the objective function (J), which is described as follows:

$$MinJ = \frac{1}{P} \sum_{i=1}^{P} \sum_{p=1}^{m} (o_p^i - d_p^i)^2 \tag{13}$$

where ${d_p}^i$ is the desired output of the *ith* input unit.

In fact, it is easy to find that FNNs training is a high-dimensional and multi-modal optimization problem, $(\omega, \theta, \hat{\omega}, \hat{\theta})$ is the decision vector, and the calculation goal is to find a set of parameters such that J is minimum. Fig. 7 depicts the basic principle of FNNs training.

4.2. Nonlinear function approximation examples

It is obvious that the superiority of the proposed paradigm is statistically significant on continuous function problems. It is also stated that CDOA is able to perform better than other algorithms for training FNNs, whose process can be treated as a challenging continuous function optimization problems characterized by missing data or noise pollution. Therefore, as suggested in literature [38], two well-studied benchmark problems (without noise and with noise) for nonlinear function approximation are required to assess the effectiveness of CDOA.

- SISO nonlinear function approximation problem:

$$o = sin(2x)exp(-2x),\ 0 \le x \le 6.2 \tag{14}$$

In this case, FNNs with the structure 1-h-1 is trained to approximate the test function. The experimental data samples from the interval [0, 6.2] with increments of 0.1, where the first 50 data is regarded as training set and the rest is the test set. Fig. 8 (a)-(b) plot this function without noise and with noise, respectively.

- MISO nonlinear function approximation problem:

$$o(t+1) = \frac{o(t)o(t-1)[o(t)+2.5]}{1+o(t)^2+o(t-1)^2} + u(t),\ u(t) = sin\left(\frac{2\pi t}{25}\right) \quad (15)$$

The simulation goal is to construct the approximation model of the non-linear function as the form below:

$$o(t+1) = F(u(t), o(t-1), o(t)) \quad (16)$$

For this problem, FNNs with the structure 3-h-1 is trained to approximation the function. The experimental data samples from the interval [0,100] with increments of 1, where the first 90 data is selected as training set and the rest is the test set. Fig. 8 (c–d) plot this function without noise and with noise, respectively.

4.3. The choice of the number of the hidden layer nodes

It is generally known that the number of hidden layer nodes (h) is an important component of FNNs. In order to determine an ideal h, the experimental data built from 2 to 10 with increments of 1 is used to investigate the influence of different h on the performance of FNNs. It can be observed from the simulation results recorded in Table 5 that the over performance of FNNs with four hidden layer nodes is better than that of other cases. According to the experiment results, FNNs with the structure 1-4-1 and 3-4-1 are therefore recommended to train SISO problem and MISO problem, respectively.

4.4. Experiments results and comparison

For solving this case, 25 independent runs are carried out on the same platform and conducted with 10,000 function evaluations as the termination criterion. Other optimization methods are also employed to solve this problem to have a valid basis of comparison with the proposed methodology. The parameters of all methods keep the same as the Section 3.3. Furthermore, the mean error value and standard deviation are applied for assessing the search capability of each approach. Table 6 summarizes the experiment results including p-value and h-value obtained by Wilcoxon's rank sum test on the corresponding statistical results.

The simulation results listed in Table 6 provide information to contend that CDOA outperforms other optimization techniques under the same termination criterion and optimization runs. Although the proposed paradigm also performs similar to other methods for some cases, the superiority is statistically significant with respective to other methods. That is, CDOA is able to achieve a set of reasonable parameter to such that the objective index is minimum. These results highly illustrate that the proposed method process demands higher accuracy in real application. Furthermore, it is also interesting that the accelerated degrade can also be observed in the convergence curves (see Fig. 9) as well, which is due the following discussed reasons. i) CDOA intrinsically benefits from high exploration and local optima

Table 6
Experiment results of all methods on SISO and MISO approximation problems.

Function	Index	CDOA	CS	CSA	FPA	PSO	ABC	GSA
SISO without noise	mean	6.0299e−04	8.0119e−04≀	1.3201e−03≀	2.1082e−03‡	2.0306e−03‡	4.3573e−04≀	3.6953e−03‡
	std	4.7957e−04	6.8744e−04	1.3466e−03	9.9428e−04	1.8312e−03	2.4376e−04	1.7228e−03
	p	–	3.7211e−01	4.0410e−01	1.6165e−07	5.9408e−04	4.4922e−01	1.3079e−08
	h	–	0	0	1	1	0	1
	Testerror	1.9849e−03	3.7695e+00‡	9.2702e−02≀	1.1012e−03≀	1.1632e−03≀	2.9968e−02≀	3.2901e−03≀
	std	7.2891e−03	5.4687e+00	4.4600e−01	2.4845e−03	2.2556e−03	8.9779e−02	8.5542e−03
	p	–	1.5967e−09	4.8486e−01	6.2762e−01	1.2061e−01	4.7281e−01	2.0034e−01
	h	–	1	0	0	0	0	0
SISO with noise	mean	5.3076e−04	1.0001e−03‡	1.7733e−03‡	1.8941e−03‡	2.9523e−03‡	6.2195e−04≀	4.1132e−03‡
	std	6.3464e−04	5.7308e−04	1.8130e−03	8.0564e−04	1.8021e−03	3.2337e−04	2.2615e−03
	p	–	4.1349e−04	5.2058e−02	4.9750e−08	1.1153e−06	5.0032e−02	4.6381e−09
	h	–	1	1	1	1	0	1
	Testerror	5.8186e−03	1.9027e+00‡	1.1171e−02≀	9.3396e−04≀	9.8418e−04≀	1.4646e−02‡	4.9323e−03†
	std	1.9988e−02	4.2813e+00	4.5265e−02	2.4718e−03	1.1440e−03	3.1866e−02	8.4730e−03
	p	–	2.5742e−09	8.0086e−01	2.1432e−01	4.9707e−01	4.3602e−02	3.1262e−02
	h	–	1	0	0	0	1	1
MISO without noise	mean	1.5956e−01	4.4524e−01‡	3.9627e−01‡	7.8157e−01‡	3.0270e−01‡	4.5262e−01‡	1.1947e+00‡
	std	6.1806e−02	1.5182e−01	2.2852e−01	1.3385e−01	1.4171e−01	1.6295e−01	2.9841e−01
	p	–	1.6401e−08	2.4151e−05	1.4157e−09	3.0245e−05	1.0585e−07	1.4157e−09
	h	–	1	1	1	1	1	1
	Testerror	2.1608e−01	3.8482e+00‡	3.9216e−01‡	7.9864e−01‡	3.3419e−01‡	6.4443e−01‡	1.7384e+00‡
	std	1.0450e−01	1.0851e+00	1.8903e−01	2.4780e−01	1.4085e−01	2.8746e−01	1.1158e+00
	p	–	1.4157e−09	4.7850e−04	6.5743e−09	2.3172e−03	3.0175e−07	1.4157e−09
	h	–	1	1	1	1	1	1
MISO with noise	mean	2.0572e−01	5.0633e−01‡	3.8686e−01‡	1.6874e−01†	4.6805e−01‡	4.8758e−01‡	1.0673e+00‡
	std	6.3185e−02	1.5766e−01	2.4013e−01	3.3284e−02	1.5552e−01	1.7107e−01	3.3839e−01
	p	–	1.4648e−08	3.1856e−03	1.8002e−09	1.9973e−06	1.4648e−08	1.4157e−09
	h	–	1	1	1	1	1	1
	Testerror	2.8022e−01	4.4123e+00‡	4.1489e−01‡	7.7156e−01‡	4.5019e−01‡	5.9005e−01‡	1.2225e+00‡
	std	9.2702e−02	2.4257e+00	2.2752e−01	3.1720e−01	2.2894e−01	3.0802e−01	6.6365e−01
	p	–	1.4157e−09	4.1620e−02	4.6381e−09	2.6346e−03	1.0108e−06	1.4648e−08
	h	–	1	1	1	1	1	1
	‡	–	7	5	5	6	5	6
	†	–	0	0	1	0	0	1
	≀	–	1	3	2	2	3	1

‡, † and ≀ indicate that the performance of CDOA is better than, worse than and similar to that of the corresponding algorithm, respectively.

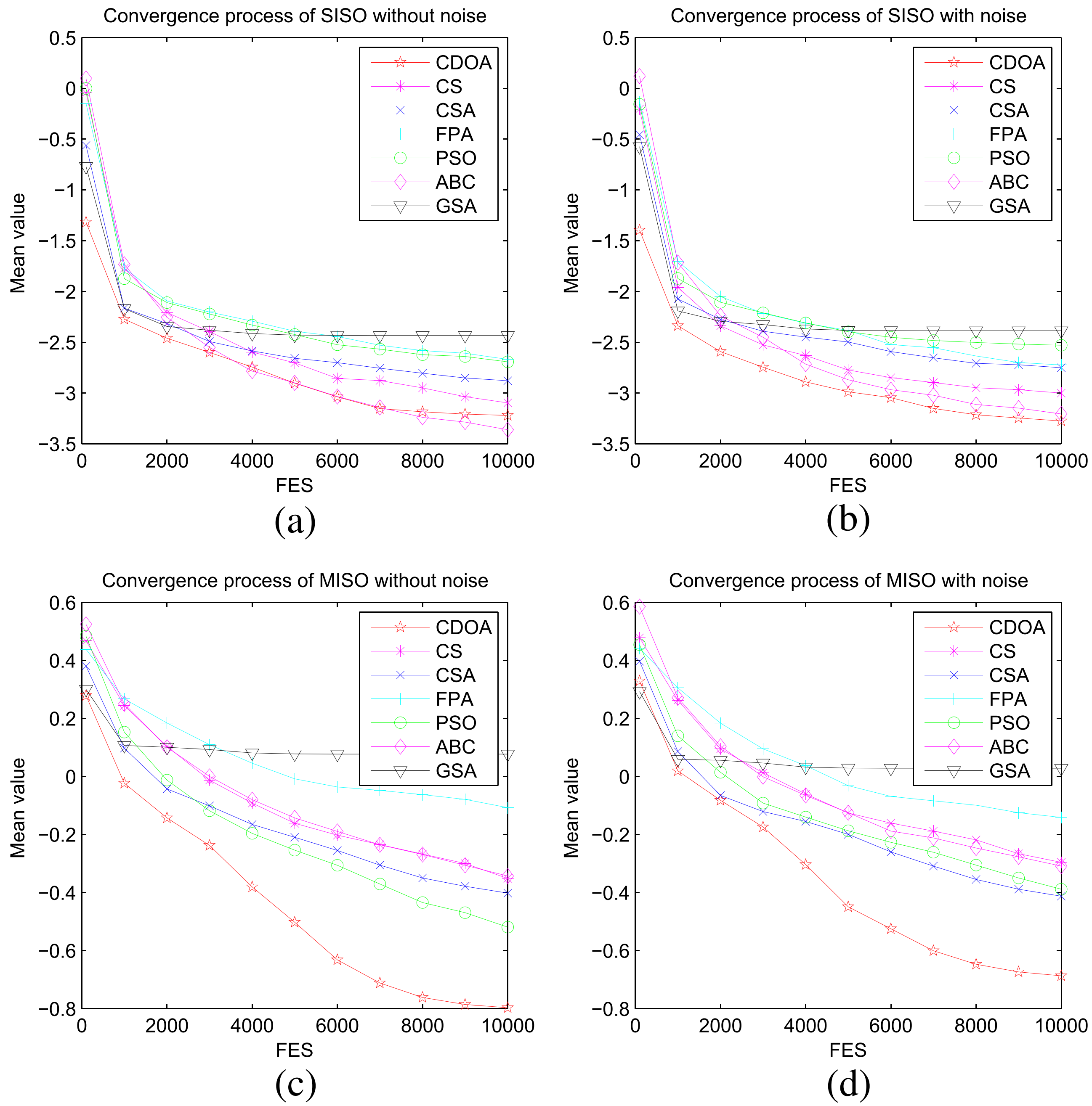


Fig. 9. The convergence curves of all methods on two approximation problems (a) SISO without noise, (b) SISO with noise, (c) MISO without noise, (d) MISO with noise.

avoidance, since it is also a population-based methodology. This guarantees the proposed approach to avoid a large amount of local solutions and explore promising areas extensively. ii) CDOA can smoothly achieve a good transformation between exploration and exploitation on the multi-step position-selected scheme, which ensures local optima avoidance at the initial steps of iteration and convergence to the most effective region in the final stages of evolution. iii) The best agent obtained so far is obliged to update its position using the local random walk and select the best one as the destination point. Hence, the candidate solutions are always improved following a tendency towards the best areas of the search spaces over the course of generations.

5. Conclusion

In this paper, the decision behavior of human is modelled to propose a new stochastic population-based method, called collective decision optimization algorithm (CDOA). The methodology is equipped with several operators to explore and exploit the solution spaces. In order to evaluate the performance of CDOA roundly, a comprehensive set of benchmark functions and training artificial neural networks are required. The extensive comparative study reveals that the proposed technique performs more effective and accurate than some of other state-of-the-art intelligent optimization methods. It can be recommended as a promising tool for optimizing different optimization tasks in various real-world fields in future work.

Acknowledgment

The authors express their sincere thanks to Dr. Seyedali Mirjalili for providing the codes. The authors would also like to thank Dr. Q.Y. Jiang and the anonymous reviewers for their free-handed assistance and construction suggestions. This work is partly supported by the National Natural Science Foundation of China under Project code (61075032), the Postdoctoral Foundation of China under Project code (2014M561817) and the Anhui Provincial Natural Science Foundation under Project code (J2014AKZR0055).

References

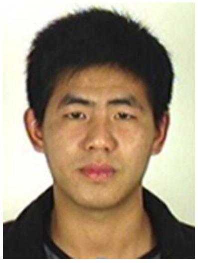
[1] X.S. Yang, Nature-inspired metaheuristic algorithms, 2nd ed., Luniver press, 2010.
[2] J.H. Holland, Genetic algorithms, Sci. Am. 267 (1992) 66–72.
[3] S. Mirjalili, S.M. Mirjalili, A. Lewis, Let a biogeograph-based optimizer train your Multi-layer perceptron, Inf. Sci. 269 (2014) 188–209.
[4] A. Askarzadeh, Bird mating optimizer: an optimization algorithm inspired by bird mating strategies, Commun. Nonlinear Sci. Numer. Simul. 19 (2014) 1213–1228.
[5] S. Kirkpatrick, C.D. Gelatt, VecchiMP. Optimization by simmulated annealing, Science 220 (1983) 671–680.
[6] A. Kaveh, M. Khayatazad, A new meta-heuristic method: ray optimization, Comput. Struct. 112 (2012) 283–294.
[7] E. Rashedi, H. Nezamabadi-Pour, S. Saryazdi, GSA: a gravitational search algorithm, Inf. Sci. 179 (2009) 2232–2248.
[8] A. Hatamlou, Black hole: a new heuristic optimization approach for data clustering, Inf. Sci. 222 (2013) 175–184.
[9] R.C. Eberhart, J. Kennedy, A new optimizer using particle swarm theory, in: Proceedings of the Sixth International Symposium on Micro Machine and Human Science, 1995, 39–43.
[10] X.S. Yang, Flower pollination algorithm for global optimization, Unconv. Comput. Nat. Comput. (2012) 240–249.
[11] X.S. Yang, S. Deb, Engineering optimization by cuckoo search, Int. J. Math. Model. Numer. Optim. 1 (2010) 330–343.
[12] A. Askarzadeh, A novel metaheuristic method for solving constrained engineering optimization problems: crow search algorithm, Comput. Struct. 169 (2016) 1–12.
[13] A.H. Gandomi, Interior search algorithm (ISA): a novel approach for global optimization, ISA Trans. 53 (2014) 1168–1183.
[14] D. Karaboga, B. Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, J. Glob. Optim. 37 (2007) 459–471.
[15] Z.D. Geem, J.H. Kim, G. Loganathan, A new heuristic optimization algorithm: harmony search, Simulation 76 (2001) 60–68.
[16] D.H. Wolpert, W.G. Macready, No free lunch theorems for optimization, IEEE Trans. Evolut. Comput. 1 (1997) 67–82.
[17] F. Gao, G. Cui, Z.B. Wu, X.Z. Yang, A novel multi-step position-selectable updating particle swarm optimization algorithm, Acta Electron. Sin. 37 (2009) 529–534.
[18] R.J. Aldag, S.R. Fuller, Beyond fiasco: a reappraisal of the groupthink phenomenon and a new model of group decision processes, Psychol. Bull. 113 (1993) 533–552.
[19] I.L. Janis, Victims of groupthink: A psychological study of foreign-policy decision and fiascoes. Boston Houghton Mifflin, 1972.
[20] A. Mohamed, F. Wiebe, Toward a process theory of groupthink, Small Group Res. 27 (1996) 416–430.
[21] A. Nahavandi, The art and science of leadership, 2006
[22] W.K. Hoy, C.J. Tarter, Swift and smart decision making: heuristics that work: heuristics that work, Int. J. Educ. Manag. 24 (2013) 351–358.
[23] K.A. McHugh F.J. Yammarino S.D. Dionne A. Serban H. Sayama S. Chatterjee, Collective decision making, leadership, and collective intelligence:Tests with agent-based simulations and a Field study, 27, 2016, 218–241
[24] S. Nitzan, J. Paroush, Collective Decision Making: an Economic Outlook, Cambridge University press, Cambridge, 1985.
[25] D. Karotkin, S. Nitzan, On two properties of the marginal contribution of individual decisional skills, Math. Soc. Sci. 34 (1997) 29–36.
[26] G. Barron, I. Erev, Small feedback-based decisions and their limited correspondence to description-based decisions, J. Behav. Decis. Mak. 16 (2003) 215–233.
[27] R.M. Hogarth, E. Soyer, Sequentially simulated outcomes: kind experience versus nontransparent description, J. Exp. Psychol.: Gen. 140 (2011) 434–463.
[28] G. Koritzky, E. Yechiam, On the robustness of description and experience based decision tasks to social desirability, J. Behav. Decis. Mak. 23 (2010) 83–99.
[29] A. Frederic, B. Patrick, C. Sven, H. Patrick, Creativity and Innovation in decision making and decision support. International conference on creativity and innovation in decision making and decision support, London, 2006.
[30] E. Gutierrez, G. Olundh Sandstrom, J. Janhager, S. Ritzen, Innovation and decision making: understanding selection and prioritization of development projects, in: proceedings of the 2008 IEEE International Conference on Management of Innovation and Technology, 2008.
[31] J. Du, H. Love James, S. Roper, The innovation decision: an economic analysis, Technovation 27 (2007) 766–773.
[32] X. Yao, Y. Liu, G. Lin, Evolutionary programming made faster, IEEE Trans. Evolut. Comput. 3 (1999) 82–102.
[33] P.N Suganthan, N. Hansen, J.J. Liang, K. Deb, et al. Problem definitions and evaluation criteria for CEC 2005 special session on real-parameter optimization. KanGAL Report, 2005.
[34] Y. Wang, Z.X. Cai, Q.F. Zhang, Differential evolution with composite trial vector generation strategies and control parameters, IEEE Trans. Evolut. Comput. 15 (2011) 55–66.
[35] A.H. Gandomi, A.H. Alavi, Krill herd: a new bio-inspired optimization algorithm, Commun. Nonlinear Sci. Numer. Simul. 17 (2012) 4831–4845.
[36] R.J. Schalkoff, Artificial Networks, McGraw-Hil, 1997.
[37] K. Homik, M. Stinchcombe, H. White, Multilayer feedforward networks are universal approximators, Neural Netw. 2 (1989) 359–366.
[38] Q.Y. Jiang, L. Wang, X.H. Hei, G.L. Yu, Y.Y. Lin, The performance comparison of a new version of artificial raindrop algorithm on global numerical optimization, Neurocomputing 179 (2016) 1–25.

Qingyang Zhang received the B.S. degree in mathematics from Beihua University, Jilin, China, in 2011, the M.S. degree in applied mathematics from Beifang University of Nationalities, Yinchuan, China, in 2015. He is currently pursuing the Ph.D. degree at the School of Computer and Information, Hefei University of Technology, Hefei, China. His current research interests include meta-heuristics, evolutionary computation and image processing.

Ronggui Wang received the M.S. degree in mathematics from Anhui University, China, in 1997, and the Ph.D. degree in computer science from Hefei University of Technology, Hefei, China, in 2005. He is currently a Professor with School of Computer and Information, Hefei University of Technology. His research interests include digital image processing, artificial intelligence and data mining.

Juan Yang received the B.S. and M.S. degrees in mathematics from Hefei University of Technology, Hefei, China, in 2004 and 2008, respectively. She received the Ph.D. degree with school of Computer and Information, Hefei University of Technology. She is currently a lecturer with school of Computer and Information, Hefei University of Technology. His research interests include image processing and intelligent visual surveillance.

Kai Ding received the B.S. degree in computer science from Hainan Normal University, China, in 2014. He is currently pursuing the M.S. degree at School of Computer and Information, Hefei University of Technology. His research interests include digital image processing, artificial intelligence and data mining.



Yongfu Li received the B.S. degrees in mathematics from Anhui university of Science and Technology, Huainan, China, in 2014. His research interests include image processing, artificial intelligence and intelligent visual surveillance.

Jiangen Hu received the B.S. degree in mathematics from Anhui architecture university, China, in 2014. He is currently pursuing the M.S. degree at School of Computer and Information, Hefei University of Technology. His research interests include Image processing, artificial intelligence and video analysis.