Journal of Ambient Intelligence and Humanized Computing
https://doi.org/10.1007/s12652-021-03234-5

ORIGINAL RESEARCH

Chaotic Coyote Optimization Algorithm

Huawei Tong[1] · Yun Zhu[2] · Juliano Pierezan[3] · Youyun Xu[1] · Leandro dos Santos Coelho[3]

Received: 24 November 2019 / Accepted: 25 March 2021

Abstract
Coyote Optimization Algorithm (COA) is classified as both swarm intelligence and evolutionary heuristic algorithms. However, getting trapped in a poor local optimum and the low convergence speed are the weaknesses of COA obviously. Due to these weaknesses, this paper proposes a new algorithm named Chaotic Coyote Optimization Algorithm (CCOA) which focusing on COA equipped with chaotic maps. Through utilising ten well-known benchmark functions, experimental results are recorded in tables and drawn in figures to provide a sharp contrast. The performance of CCOA and COA are discussed, which proves CCOA outperforms COA guaranteeing rapid global convergence rate.

Keywords Coyote Optimization Algorithm · Chaos theory · Swarm intelligence algorithms · Optimization

1 Introduction

Many scholars are trying to solve optimization problems in their areas of expertise through various methods, such as the modulation orders on the LEDs in visible light communication (Wang et al. 2018a, b). One of the effective means is the meta-heuristic optimization algorithm. The power and efficiency of the performance of nature-inspired meta-heuristic algorithms make it suitable for dealing with high-dimension nonlinear optimization problems. For the most part, nature-inspired meta-heuristic algorithms are regarded as approaches to maintain a balance between intensification (local search) and randomization (global search) (Wang et al. 2014). Around the current best solutions, the task involves searches and selects the best candidate designs achieved by intensification searches. Randomization generally refers to diversification, which endows the optimizer with higher exploration capability ability in the search space (Gandomi et al. 2013b).

The most famous algorithms are the ant colony algorithm and the particle swarm algorithm, which have improved their performance over the years and have a rich variety of application scenarios and mature application solutions. The former (Dorigo and Caro 1999; Dorigo et al. 2006) is based on simulations of the habits of ants to provide a new approach to the optimization problem and to point out a new direction for optimization research. The latter has been widely studied (Kennedy and Eberhart 1995; Eberhart and Kennedy 1995; Clerc and Kennedy 2002), which is based on the study of the foraging behavior of a flock of birds, his in-depth study of learning cooperation among individuals, an in-depth study of the learning cooperation between individuals, and a reliable algorithmic framework and a feasible process are summarized through mathematical modeling analysis. Many new meta-heuristic algorithms have been proposed, while classical algorithms are still being effectively improved by researchers.

Based on white hole, black hole, and wormhole, which are concept in cosmology, multi-verse optimizer (MVO) (Mirjalili et al. 2016) is proposed to apply to engineering problems, the results demonstrate MVO has great potential to solve practical problems with unknown search spaces. In order to Achieve a good balance between exploration and

✉ Huawei Tong
1018010108@njupt.edu.cn

Yun Zhu
yzhu@gnnu.cn

Juliano Pierezan
juliano.pierezan@ufpr.br

Youyun Xu
yyxu@njupt.edu.cn

Leandro dos Santos Coelho
leandro.coelho@pucpr.br

[1] Nanjing University of Posts and Telecommunications, Nanjing 210003, China

[2] Gannan Normal University, Ganzhou 341000, China

[3] Federal University of Parana (UFPR), Curitiba, Brazil

Published online: 13 April 2021

exploitation, a novel algorithm named Rat swarm optimizer (RSO) (Dhiman et al. 2020) is developed by observing the chasing and attacking behaviors of rats. Inspired by the social behavior of humpback whales, a new algorithm called the whale optimization algorithm (WAO) provide by Mirjalili and Lewis (2016). And it has been proved that WOA has advantages that neither the state-of-art meta-heuristic algorithms nor classical algorithms can match. There are also some recent and popular algorithms, such as Salp swarm algorithm (Mirjalili et al. 2017) and Henry gas solubility optimization (Hashim et al. 2019).

While improving the meta-heuristic algorithm, researchers also apply the algorithm to a specific scenario to solve some practical problems. Artificial bee colony algorithm (ABC) has been adapted to handle the design of linear phase two-channel quadrature mirror filter (QMF) banks (Agrawal and Sahu 2015). In terms of measure of ripple, mean square in pass-band and stop-band edge attenuation, the performance of the proposed method is significantly better than other methods for the design problem of N = 48. A mixture of culture algorithm and heritage mechanism, ABC was successfully improved and its application scenario is wireless sensor networks (Saad et al. 2019). The PSO algorithm is introduced into the Salp Swarm Algorithm, and a highly efficient SSAPSO algorithm for is proposed for feature extraction which has become a research hotspot for data mining and machine learning. SSAPSO overcomes the shortcomings of SSA such as slow convergence (Ibrahim et al. 2019). PSO also has a number of logical and clever improvements that can be applied to cloud computing (Naseri and Jafari Navimipour 2019; Mansouri et al. 2019).

The optimization research disciplines combine chaos theory has attracted a great deal of attentions last few years. The properties that randomness and ergodicity are the inherit characteristics of chaotic maps (Sayed et al. 2019). Obviously, utilizing chaos theory can substitute the vital role of randomizing. The measure that the random elements replaced by chaotic elements will enhance the swarm optimization algorithm confirmed in the paper (Jiang 1998).

Among the algorithms utilizing chaos theory, Chaotic Krill Herd algorithm (CKH) (Wang et al. 2014) has better results than Krill Herd algorithm (KH), Chaotic biogeography-based optimisation (CBBO) (Saremi et al. 2014) has enhanced the performance of biogeography-based optimisation (BBO), chaotic operator improve the ability of algorithms to solve some practical problems in (Gwo-Ching and Ta-Peng 2006; LüQ and Yu 2003; Jia 2009), Chaotic crow search algorithm (CCSA) is superior compared to crow search algorithm (CSA) (Sayed et al. 2019), the capacity of Chaotic particle swarm optimization algorithm (CPSO) is higher than PSO (Jinfeng 2011), the global convergence speed of Chaotic whale optimization algorithm (CWOA) has been enhanced by chaos theory (Kaur and Arora 2018), Chaotic GWO (CGWO) algorithm (Kohli and Arora 2018), Chaotic improved firefly algorithms (CFAs) (Gandomi et al. 2013a), and so on.

The rest of the paper is organized as follows: Sect. 2 presents an introduction to the Coyote Optimization Algorithm (COA). Section 3 describes the new algorithm Chaotic Coyote Optimization Algorithm (CCOA) which combines with chaotic maps. Section 4 tests new algorithms with benchmark functions, and keeps a record of all the experimental results. Section 5 concludes this paper.

2 Coyote Optimization Algorithm

Like other algorithms, Coyote Optimization Algorithm (COA) is inspired on the species called Canis latrans. Pierezan and Coelho (2018) states that COA has obtained the best average ranking in most cases than the other swarm algorithms, like artificial bee colony (ABC) (Karaboga and Basturk 2007), the firefly algorithm (FA) (Yang 2010), etc.

The COA focus on the coyotes' social conditions SoC, which can be calculated as follows

$$SoC_{coy}^{pa,t} = \vec{S} = \left(S_1, S_2, \ldots, S_D\right), \tag{1}$$

where $\vec{S}$ is the decision variables of a global optimization problem, the search space dimension of that problem is D, pa is the serial number of the pack, and coy is the serial number of the coyote that in this pack. t represents the instant of time when the social condition is calculated.

The social conditions of COA include initialization and update, the former is calculated by the formula as follows

$$SoC_{coy,d}^{pa,t} = L_d + r_d \times \left(U_d - L_d\right), \tag{2}$$

where L_d and U_d mean separately the lower and upper bounds of the dth decision variable. The number r_d in range [0,1] is generated casually.

Inside the packs, the detailed analysis of the cultural interaction has several limitations. Under the influence of the alpha (λ_1) is a restriction on all coyotes, that means a cultural difference from a random coyote of the pack ($coyote_1$) to the alpha coyote. And another restriction is that coyotes must not be over the influence of the pack (λ_2), which means a cultural difference from a random coyote ($coyote_2$) to the cultural tendency of the pack. (λ_1) and (λ_2) are defined separately as

$$\lambda_1 = alpha^{pa,t} - SoC_{coy_1}^{pa,t} \tag{3}$$

$$\lambda_2 = culture^{pa,t} - SoC_{coy_2}^{pa,t}. \tag{4}$$

In the process of the algorithm, the social conditions are updated by the rule as follows

$$NewSoC_{coy}^{pa,t} = SoC_{coy}^{pa,t} + r_1 \times \lambda_1 + r_2 \times \lambda_2. \tag{5}$$

In this rule, r_1 and r_2 are the random numbers in the range [0,1], and they represent the weights of the alpha and the influence of pack.

The main step of the application of the COA for solving optimization problem is to evaluate the adaptation of coyotes in the respective current social conditions.

$$Fit_{coy}^{pa,t} = FitFunc\left(SoC_{coy}^{pa,t}\right) \tag{6}$$

$$NewFit_{coy}^{pa,t} = FitFunc\left(NewSoC_{coy}^{pa,t}\right), \tag{7}$$

where *FitFunc* is the fitness function.

Algorithm 1 Coyote Optimization Algorithm

Require:
The numbers of the coyotes: $NumCoy$
The numbers of packs: $NumPac$
The fitness function: $FitFunc$
The maximum number of the coyotes: $MaxNumCoy$
Ensure: The global minimum cost
1: Initialize the social conditions of all coyotes (Eq. 2).
2: Calculate the costs of all coyotes (Eq. 6).
3: sort costs in ascending order.
4: **if** $NumCoy < MaxNumCoy$ **then**
5: **for** each *coy* coyote of the *pa* pack **do**
6: Update the social conditions of coyotes (Eq. 5).
7: Update the costs of coyotes (Eq. 7).
8: **if** a random number $RandNum < 0.005 \times NumCoy^2$ **then**
9: The coyote: *coy* leaves its packs and becomes lonely or enter other packs.
10: **end if**
11: Sort costs in ascending order.
12: **end for**
13: **end if**
14: **return** $GlobalMinCost$

3 Chaotic Coyote Optimization Algorithm

Fortunately, the weaknesses of COA that trapped in the poor local optimum and the low convergence speed, can be overcomed by some effective methods, including chaotic theory. In this section, chaotic maps have been employed to improve the performance of COA. That is to say, the modifications have been done in the following to strengthen the performance and the robustness of the COA. The new algorithm derived from COA with chaotic operatiors is called CCOA.

3.1 Chaotic maps for initializing social conditions

The chaotic matrix is produced to instead the random number r_d which can be seen in Eq. (2) by using chaotic maps for initializing the social conditions. Therefore, Eq. (2) should be rewritten as follows

$$SoC_{coy,d}^{pa,t} = L_d + Chaotic_d \times \left(U_d - L_d\right). \tag{8}$$

3.2 Chaotic maps for updating social condition

Two sequences are produce by chaotic maps for updating the social conditions of the coyotes. Hence, Eq. (5) is rewritten as follows

$$NewSoC_{coy}^{pa,t} = SoC_{coy}^{pa,t} + Chaotic_1 \times \lambda_1 + Chaotic_2 \times \lambda_2. \tag{9}$$

3.3 Chaotic maps for the migration of coyotes

The real random number replaced by a chaotic number replaces in the Line 8 of Algorithm 1. Then the sentence is changed as follows

If a chaotic number $ChaoticNum < 0.005 \times NumCoy^2$ then

3.4 Chaotic maps for initializing social conditions, updating social condition, and the migration of coyotes

The usages of chaotic maps above all shared in this case. That means the random numbers in Eqs. (2, 5), and Line 8 of Algorithm 1, are superseded by deterministic chaotic signals.

Table 1 The benchmark functions and chaotic maps used in CCOA

		Benchmark function	Chaotic map
NAME	1	Ackley	Logistic
	2	Sphere	Tent
	3	Shifted sphere	Iterative
	4	Salomon	Gauss/mouse
	5	Shifted Rosenbrock	Piecewise
	6	Rosenbrock	Sine
	7	Rastrigin	Singer
	8	Penalty 1	Sinusoidal
	9	Penalty 2	Chebyshev
	10	Schwefel	Circle

Table 2 The set of parameters of Chaotic Coyote Optimization Algorithm

Parameter	Value
The number of coyotes in each pack	5
The number of packs	20
Experiment times	30
Initial value of the chaotic maps for initializing social conditions	0.8
Initial value of the chaotic maps for updating social conditions	0.8, 0.3
Initial value of the chaotic maps for migration social conditions	0.8
The maximum number of the coyotes	200000

4 Simulation and results

For explaining clearly in the comparison, COA with using chaotic maps in the four cases are called separately CCOA1-CCOA10 (for initializing social conditions), CCOA11-CCOA20 (for updating social conditions), CCOA21-CCOA30 (for the migration of coyotes), CCOA31-CCOA40 (for initializing social conditions, updating social condition, and the migration of coyotes) in this section. Here, the definition of the ten chaotic maps used in this algorithm are listed in Wang et al. (2014), and their figures are drawn in Saremi et al. (2014). As benchmark suite, ten test problems selected that can be found in the CEC'2005 special session (Suganthan et al. 2005). Meanwhile the use of ten benchmark functions in comparing CCOA1-CCOA40 and COA, whose differences will be contrasted through the use of nonparametric procedures. Table 1 shows the name of benchmark functions and chaotic maps used in this article. Each algorithm recorded in Tables 3, 4, 5, 6 in order uses the chaotic map listed in Table 1 individually. In addition, the important parameters are also displayed in the Table 2.

It is clear that the rang of iterative map and Chebyshev map need to be normalized. The formula is used as follows

$$x_{normalization}(i) = \frac{x(i) - min(x)}{max(x) - min(x)}, \tag{10}$$

where $x(i)$ is the chaotic sequence produced by Iterative map or Chebyshev map, $min(x)$ is the minimum number of the sequence, $max(x)$ is the maximum number of the sequence.

To begin with setting a series of parameters for experiment, then the mean, standard deviation (SD) and p value of tested subjects are measured respectively. Learning from (Derrac et al. 2011), doing a most convincing persuasive work about describing the performance of an algorithm must include statistical test. The efficiency of Wilcoxon signed-rank test is robustness (Vorapratee p 2013). Wilcoxon's rank-sum test with the 0.95 confidence level is suitable utilized in the experiment. p value as mentioned above is calculated by using the Wilcoxon rank-sum, which has been marked a font bold as well as underlined while the value over 0.05. When $p > 0.05$, there is no significant difference between algorithms. In Tables 3, 4, 5, 6, *NaN* signifies that two algorithms can not be compared.

4.1 Comparison of COA with CCOA

Table 3 displays the results of CCOA algorithms with chaotic initializing operators. As this table shows, none of the algorithms has the absolute superiority over others. This completion also can be obtained from Fig. 1 identically, where sorting all algorithms into ascending order according to the data *MEAN* in Table 3. In other words, the higher performance ranking of the vertical coordinate, the faster convergence of the algorithm. Algorithms CCOA1-10 do not succeed to provide significant superiority confirmed by the p value in Table 3. For example, CCOA10 overtakes all other algorithms on function 1, 3, and 6, while CCOA5 has the best performance on function 2 and 8. Whereas CCOA2, CCOA3, CCOA7, CCOA8, CCOA9 provide better results on function 4, 5, 7, 9, and 10 separately. The convergence curves of all algorithms crowd together in almost the whole optimization process shown in Fig. 2.

Table 4 illustrates the values of CCOA algorithms with chaotic updating operators. From Fig. 4, apparently, other algorithms are outperformed by CCOA18 on the benchmark funtcion 1, 2, 3, 7, and 8. Meanwhile CCOA11, CCOA13, CCOA17 provide better results on function 9, 10, and 5 respectively. As shown in Fig. 3, the algorithms except CCOA14 and CCOA15 converge to the closest value at last. However, all algorithms show worse results on function 4, 6 shown in Fig. 4, and CCOA20 failed on all test functions.

Table 5 shows the values obtained with ten chaotic maps on all benchmark functions. The algorithms include CCOA24, CCOA25, CCOA26, CCOA29, and CCOA30, perform better than COA and other algorithms. Figure 5

Table 3 The result of use of chaotic maps for initializing social condition

Function	Algorithms										
	COA	CCOA1	CCOA2	CCOA3	CCOA4	CCOA5	CCOA6	CCOA7	CCOA8	CCOA9	CCOA10
Ackley											
MEAN	7.576376E–08	8.585365E–08	5.648769E–08	8.341110E–08	1.746920E–07	5.811548E–08	1.775989E–07	5.566622E–08	4.791464E–08	9.066658E–08	3.389078E–08[a]
SD	2.355940E–07	1.506580E–07	1.432527E–07	2.477635E–07	2.873849E–07	1.602679E–07	5.232366E–07	1.459545E–07	1.213046E–07	1.276927E–07	8.318456E–08[b]
p	1.000000	3.319196E–11	6.551578E–12	9.102737E–12	2.164832E–11	1.406492E–11	4.104007E–11	3.319196E–11	2.411006E–12	2.164832E–11	2.409815E–11
Sphere											
MEAN	6.324382E–13	1.056196E–13	2.925099E–13	2.266796E–14	1.479567E–12	6.917936E–15[a]	3.696430E–12	4.864824E–14	1.049727E–12	9.306178E–13	1.373358E–14
SD	2.765086E–12	4.505423E–13	1.266128E–12	9.708303E–14	6.470945E–12	2.917030E–14[b]	1.617023E–11	1.934200E–13	4.584243E–12	8.852794E–12	5.454540E–14
p	1.000000	3.768799E–12	1.944285E–11	2.164832E–11	2.409815E–11	2.154937E–12	1.925598E–12	2.696851E–12	1.944285E–11	2.164832E–11	2.681873E–11
Shifted sphere											
MEAN	8.934043E–14	2.330391E–14	3.209458E–14	7.111659E–15	8.172181E–14	1.603171E–13	3.303005E–13	5.046926E–15	2.907110E–13	1.497485E–14	3.558040E–15[a]
SD	3.513040E–13	9.970369E–14	1.398062E–13	2.781581E–14	3.516565E–13	7.010382E–13	1.285471E–12	1.771182E–14	1.264493E–12	5.629325E–14	1.641600E–14[b]
p	1.000000	2.681873E–11	5.868503E–12	3.371781E–12	5.868503E–12	1.132047E–11	7.312394E–12	4.561823E–11	5.868503E–12	2.154937E–12	5.868503E–12
Salomon											
MEAN	0.117014	0.115967	0.108262[a]	0.117747	0.122300	0.109474	0.121095	0.111911	0.117194	0.114815	0.129254
SD	0.010630	0.006974[b]	0.008947	0.015153	0.014754	0.007230	0.013545	0.011916	0.010792	0.014980	0.013387
p	1.000000	7.362573E–12	1.509915E–12	1.509915E–12	1.509915E–12	1.509915E–12	1.509915E–12	1.509915E–12	1.509915E–12	1.509915E–12	7.362573E–12
Shifted Rosenbrock											
MEAN	4.514106	2.911106	3.399913	1.464804[a]	1.601749	3.012024	3.708792	3.816927	3.246600	2.160707	4.001709
SD	0.941888	0.727627	0.627163	0.354557[b]	0.656742	0.646480	1.002732	0.742019	1.020826	1.048516	0.693578
p	1.000000	9.102737E–12	2.154937E–12	2.154937E–12	3.015857E–12	1.261983E–11	4.211546E–12	7.312394E–12	9.102737E–12	1.132047E–11	1.406492E–11
Rosenbrock											
MEAN	1.787732	2.308165	2.865477	5.401552	1.854576	2.418292	2.435647	1.378389	3.796268	3.619689	0.908651[a]
SD	0.547281	0.616260	0.660600	1.144939	0.444662	0.380898	0.480278	0.413836	0.928765	0.691116	0.282446[b]
p	1.000000	1.567171E–11	1.567171E–11	4.104007E–11	8.159589E–12	5.255375E–12	8.159589E–12	2.164832E–11	4.561823E–11	1.015245E–11	4.705168E–12
Rastrigin											

Table 3 (continued)

Function	Algorithms										
	COA	CCOA1	CCOA2	CCOA3	CCOA4	CCOA5	CCOA6	CCOA7	CCOA8	CCOA9	CCOA10
MEAN	1.475385E–12	8.837778E–13	3.832824E–12	5.551234E–13	2.449096E–11	4.917382E–13	3.831494E–11	4.630323E–13[a]	1.860258E–11	6.189396E–13	3.633640E–12
SD	5.872915E–12	3.491938E–12	1.621156E–11	2.080266E–12	1.072440E–10	1.750563E–12	1.676285E–10	1.717543E–12[b]	8.130733E–11	2.537626E–12	1.906100E–11
p	NaN	NaN	0.333711[c]	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Penalty 1											
MEAN	2.590853E–15	2.404750E–15	9.363300E–15	1.318674E–13	2.646004E–13	1.999075E–15[a]	2.628803E–15	9.618282E–15	1.009097E–14	4.730475E–13	5.452586E–15
SD	9.673585E–15	1.028524E–14	6.212496E–14	5.542676E–13	1.226941E–12	7.672694E–15[b]	1.068249E–14	4.106064E–14	4.447226E–14	4.388058E–12	2.382346E–14
p	1.000000	1.406492E–11	1.406492E–11	2.696851E–12	2.983925E–11	7.312394E–12	1.132047E–11	5.868503E–12	2.696851E–12	2.154937E–12	2.154937E–12
Penalty 2											
MEAN	6.066838E–14	5.467167E–14	4.471060E–12	2.432280E–13	4.047434E–13	2.237957E–14	1.686358E–12	1.125252E–13	9.855239E–15[a]	5.875010E–13	7.804868E–14
SD	2.605594E–13	2.156504E–13	1.958055E–11	1.074182E–12	1.718618E–12	9.303227E–14	7.484180E–12	4.624513E–13	3.403016E–14[b]	2.567705E–12	3.515717E–13
p	1.000000	2.164832E–11	9.102737E–12	8.159589E–12	2.164832E–11	2.409815E–11	9.102737E–12	1.567171E–11	1.720251E–12	2.164832E–11	5.868503E–12
Schwefel											
MEAN	1.272757E–04	1.272761E–04	1.272757E–04	1.272757E–04	1.301030E–04	1.272757E–04	1.272757E–04	1.272757E–04	1.272757E–04	1.272757E–04[a]	1.272757E–04
SD	1.025796E–10	1.807666E–09	6.654262E–11	2.554249E–12	1.238618E–05	3.160182E–12	3.005304E–10	8.234659E–12	2.389581E–12	1.893899E–12[b]	5.993459E–12
p	1.000000	3.520262E–13	3.520262E–13	3.520262E–13	3.520262E–13	3.520262E–13	3.520262E–13	3.520262E–13	3.520262E–13	3.520262E–13	3.520262E–13

[a] the minimum MEAN value, [b] the minimum SD value, [c] p value over 0.05

Table 4 The use of chaotic maps for updating social condition

Function	Algorithms										
	COA	COA11	COA12	COA13	COA14	COA15	COA16	COA17	COA18	COA19	COA20
Ackley											
MEAN	7.576376E–08	5.631463E–08	4.510011E–06	7.625685E–08	1.399595E–04	6.099775E–06	4.828240E–08	2.388965E–08	2.140001E–08[a]	5.076221E–07	1.027389E–07
SD	2.355940E–07	2.052506E–07	1.856719E–05	2.577891E–07	2.345428E–04	8.161686E–06	1.086246E–07	8.387714E–08	7.823631E–08[b]	2.017630E–06	2.629998E–07
p	1.000000	2.411006E–12	4.561823E–11	4.705168E–12	4.561823E–11	4.561823E–11	6.551578E–12	3.768799E–12	2.154937E–12	2.164832E–11	3.319196E–11
Sphere											
MEAN	6.324382E–13	1.766125E–14	1.416982E–12	5.087690E–14	4.075430E–08	1.383534E–10	1.677211E–14	2.372259E–13	4.772902E–16[a]	3.105371E–14	5.041926E–14
SD	2.765086E–12	7.417198E–14	5.891383E–12	2.060459E–13	1.379507E–07	4.891076E–10	6.042033E–14	1.038636E–12	2.037350E–15[b]	1.087698E–13	1.735706E–13
p	1.000000	4.705168E–12	4.561823E–11	4.705168E–12	4.561823E–11	4.561823E–11	2.164832E–11	2.154937E–12	1.720251E–12	3.319196E–11	3.768799E–12
Shifted sphere											
MEAN	8.934043E–14	2.404779E–13	1.749158E–10	3.870705E–13	6.122966E–08	2.265634E–09	5.303564E–14	1.038914E–13	1.348402E–15[a]	1.376910E–12	2.603751E–13
SD	3.513040E–13	1.052732E–12	7.654531E–10	1.598730E–12	2.038182E–07	1.101685E–08	2.237064E–13	4.547701E–13	6.160230E–15[b]	8.307423E–12	1.104742E–12
p	1.000000	2.154937E–12	4.561823E–11	1.925598E–12	4.561823E–11	4.561823E–11	2.409815E–11	3.371781E–12	1.720251E–12	4.561823E–11	1.745784E–11
Salomon											
MEAN	0.117014[a]	0.123329	0.146574	0.136695	0.322941	0.211597	0.123405	0.132020	0.140931	0.147488	0.119667
SD	0.010630	0.011942	0.026186	0.009295[b]	0.048055	0.011255	0.011003	0.012084	0.011844	0.026967	0.012805
p	1.000000	9.851433E–12	1.230556E–11	1.509915E–12	1.714677E–11	7.362573E–12	1.509915E–12	1.371651E–11	1.371651E–11	1.509915E–12	1.509915E–12
Shifted Rosenbrock											
MEAN	4.514106	1.970144	4.569896	4.792607	25.625105	16.746200	1.673901	0.998794[a]	2.042619	2.572053	3.513681
SD	8.915598E–04	1.401229	0.815097	1.376562	7.521212	1.861963	1.347788	0.505686[b]	0.862009	1.168139	0.959340
p	1.000000	2.164832E–11	3.319196E–11	1.745784E–11	2.409815E–11	4.561823E–11	2.411006E–12	3.371781E–12	1.720251E–12	1.132047E–11	9.102737E–12
Rosenbrock											
MEAN	1.787732[a]	3.089579	3.912491	5.106545	24.905231	9.143625	3.377614	3.947078	2.196963	5.718428	6.226873
SD	0.547281	0.663281	0.755875	0.944296	7.099721	1.405390	1.224005	1.457573	0.434106[b]	0.762701	1.323147
p	1.000000	8.159589E–12	3.691245E–11	8.159589E–12	4.104007E–11	2.409815E–11	2.681873E–11	2.409815E–11	2.696851E–12	5.255375E–12	1.261983E–11
Rastrigin											

Table 4 (continued)

Function	Algorithms										
	COA	COA11	COA12	COA13	COA14	COA15	COA16	COA17	COA18	COA19	COA20
MEAN	1.475385E–12	1.244066E–12	2.748177E–10	2.086343E–11	6.955808E–06	6.617031E–08	9.061457E–13	1.515237E–12	1.550641E–13[a]	5.332168E–10	2.142854E–10
SD	5.872915E–12	4.069618E–12	1.121012E–09	9.984774E–11	2.970753E–05	2.837767E–07	3.244005E–12	6.530135E–12	6.101168E–13[b]	2.318461E–09	7.847471E–10
p	NaN	NaN	0.333711	NaN	0.333711	0.333711	NaN	NaN	NaN	0.333711	NaN
Penalty 1											
MEAN	2.590853E–15	2.675791E–15	1.064113E–12	1.915371E–15	5.080835E–09	2.907903E–11	4.453643E–14	7.677646E–16	7.794052E–17[a]	1.698137E–14	3.184580E–15
SD	9.673585E–15	1.166189E–14	4.601182E–12	8.101457E–15	1.870420E–08	1.179671E–10	1.924946E–13	3.339531E–15	3.777514E–16[b]	6.739243E–14	2.017124E–14
p	1.000000	3.015857E–12	4.104007E–11	1.132047E–11	4.561823E–11	4.561823E–11	7.312394E–12	1.925598E–12	4.211546E–12	4.561823E–11	2.409815E–11
Penalty 2											
MEAN	6.066838E–14	3.912462E–15	5.677100E–12	8.366410E–14	7.294430E–08	1.374980E–09	1.028470E–14	7.018690E–15	8.648946E–16[a]	4.675769E–11	2.866257E–10
SD	2.605594E–13	1.553645E–14	2.083371E–11	3.377485E–13	3.049301E–07	5.985074E–09	3.762344E–14	2.803414E–14	3.128236E–15[b]	2.043646E–10	1.255268E–09
p	1.000000	3.015857E–12	4.561823E–11	3.768799E–12	4.561823E–11	4.561823E–11	2.164832E–11	2.696851E–12	1.720251E–12	8.159589E–12	2.164832E–11
Schwefel											
MEAN	1.272757E–04	1.272757E–04	1.272757E–04	1.272757E–04	1.275483E–04	1.285786E–04	1.272757E–04[a]	1.272757E–04	1.272757E–04	1.272757E–04	1.272757E–04
SD	1.025796E–10	2.198222E–10	5.325031E–11	1.202814E–11	8.701894E–07	5.704158E–06	3.600410E–12[b]	2.767954E–10	4.794044E–12	1.559743E–10	9.589068E–12
p	1.000000	3.018285E–12	3.018285E–12	3.520262E–13	3.619766E–12	3.619766E–12	3.520262E–13	3.520262E–13	3.018285E–12	3.018285E–12	3.018285E–12

[a] the minimum MEAN value, [b] the minimum SD value

Table 5 The use of chaotic maps for the migration of coyotes

Function	Algorithms										
	COA	CCOA21	CCOA22	CCOA23	CCOA24	CCOA25	CCOA26	CCOA27	CCOA28	CCOA29	CCOA30
Ackley											
MEAN	7.576376E–08	1.897127	1.839037	1.922987	2.754419E–14[a]	2.409439E–13	6.872677E–11	1.956495	1.913992	2.097416E–13	1.004406E–10
SD	2.355940E–07	0.835560	0.836544	0.836861	8.618609E–15[b]	1.166903E–13	5.045037E–11	0.873105	0.813628	1.010644E–13	2.174734E–10
p	1.000000	4.561823E–11	4.561823E–11	4.561823E–11	1.720251E–12	1.720251E–12	1.720251E–12	4.561823E–11	4.561823E–11	1.720251E–12	1.720251E–12
Sphere											
MEAN	6.324382E–13	2.367164	1.877912	1.988567	4.714902E–27[a]	3.169458E–25	8.361760E–20	2.529668	2.260158	2.115717E–25	3.158051E–20
SD	2.765086E–12	4.962818	3.915556	4.218291	3.121265E–27[b]	3.367637E–25	3.167470E–19	5.178823	4.597222	2.617539E–25	1.073503E–19
p	1.000000	4.561823E–11	4.561823E–11	4.561823E–11	1.720251E–12	1.720251E–12	1.720251E–12	4.561823E–11	4.561823E–11	1.720251E–12	1.720251E–12
Shifted sphere											
MEAN	8.934043E–14	2.392597	1.738740	1.978596	4.992870E–27[a]	4.521151E–25	4.882464E–20	2.176384	2.129845	5.380616E–25	2.926308E–19
SD	3.513040E–13	4.584586	3.418106	4.333602	4.480924E–27[b]	7.437751E–25	1.785975E–19	5.018936	4.633508	9.006965E–25	1.200865E–18
p	1.000000	4.561823E–11	4.561823E–11	4.561823E–11	1.720251E–12	1.720251E–12	1.720251E–12	4.561823E–11	4.561823E–11	1.720251E–12	1.720251E–12
Salomon											
MEAN	0.113687	1.293687	1.317162	1.280993	0.099873[a]	0.106540	0.106540	1.289263	1.309927	0.099873	1.098734E–01
SD	9.058847E–03	0.411254	0.397113	0.383074	4.172353E–09	1.004127E–09	8.717895E–08	0.414397	0.414758	6.046273E–14[b]	7.014953E–07
p	1.000000	6.404564E–12	6.404564E–12	6.404564E–12	6.157899E–13	6.157899E–13	5.695027E–12	6.404564E–12	6.404564E–12	6.157899E–13	6.157899E–13
Shifted Rosenbrock											
MEAN	4.514106	7.404456E+03	5.969358E+03	8.126379E+03	0.371913[a]	0.648779	1.273235	7.344735E+03	9.484233E+03	1.255349	1.430358
SD	8.915598E–04	1.286286E+04	1.095041E+04	1.713839E+04	0.017971[b]	0.046802	0.212827	1.492773E+04	2.207444E+04	0.081625	0.282544
p	1.000000	3.691245E–11	4.561823E–11	2.409815E–11	1.720251E–12	1.720251E–12	8.159589E–12	3.319196E–11	4.561823E–11	1.745784E–11	6.551578E–12
Rosenbrock											
MEAN	3.312260	5.869976E+03	6.425073E+03	7.384592E+03	0.144151[a]	0.179056	1.260433	6.465930E+03	6.421676E+03	0.272278	0.554597
SD	0.845421	1.105506E+04	1.266858E+04	1.737678E+04	0.003020[b]	0.010737	0.131655	1.329260E+04	1.404611E+04	0.006361	0.196408

Table 5 (continued)

Function	Algorithms										
	COA	CCOA21	CCOA22	CCOA23	CCOA24	CCOA25	CCOA26	CCOA27	CCOA28	CCOA29	CCOA30
p	1.000000	3.319196E–11	3.319196E–11	4.561823E–11	1.925598E–12	1.720251E–12	3.371781E–12	4.561823E–11	3.319196E–11	3.371781E–12	3.768799E–12
Rastrigin											
MEAN	1.475385E–12	2.251307	2.176895	2.238193	0.000000[a]	0.000000	0.000000	2.282107	2.300282	0.000000	0.000000
SD	5.872915E–12	1.242252	1.300608	1.369595	0.000000[b]	0.000000	0.000000	1.260634	1.259100	0.000000	2.801508E–15
p	NaN	0.333711	0.333711	0.333711	NaN	NaN	NaN	0.333711	0.333711	NaN	NaN
Penalty 1											
MEAN	2.590853E–15	0.126557	0.148942	0.149012	9.184966E–29[a]	2.600686E–26	3.027187E–21	0.151753	0.138593	1.496036E–26	3.415328E–21
SD	9.673585E–15	0.212717	0.277413	0.258019	8.046186E–29[b]	2.220539E–26	1.087542E–20	0.283263	0.256957	1.483367E–26	1.146173E–20
p	1.000000	4.561823E–11	4.561823E–11	4.561823E–11	1.720251E–12	1.720251E–12	1.720251E–12	4.561823E–11	4.561823E–11	1.720251E–12	1.720251E–12
Penalty 2											
MEAN	6.066838E–14	0.250824	0.255435	0.225436	1.156136E–27[a]	1.113860E–25	1.794671E–20	0.238678	0.254360	1.720898E–25	3.123496E–20
SD	2.605594E–13	0.379719	0.388499	0.312099	8.028382E–28[b]	1.112521E–25	4.855854E–20	0.329336	0.352827	1.853562E–25	1.015416E–19
p	1.000000	4.561823E–11	4.561823E–11	4.561823E–11	1.720251E–12	1.720251E–12	1.720251E–12	4.561823E–11	4.561823E–11	1.720251E–12	1.720251E–12
Schwefel											
MEAN	1.272757E–04	21.887556	22.520062	21.275171	1.272757E–04[a]	1.272757E–04	1.272757E–04	22.014092	24.361444	1.272757E–04	1.272757E–04
SD	1.025796E–10	33.150481	34.318948	30.624282	0.000000[b]	0.000000	1.017358E–13	32.547549	34.294878	0.000000	6.593936E–14
p	1.000000	3.619766E–12	3.619766E–12	3.619766E–12	3.520262E–13	3.520262E–13	3.520262E–13	3.619766E–12	3.619766E–12	3.520262E–13	3.520262E–13

[a] the minimum MEAN value, [b] the minimum SD value

Table 6 The use of chaotic maps for initializing social conditions, updating social condition, and the migration of coyotes

Function	Algorithms										
	COA	CCOA31	CCOA32	CCOA33	CCOA34	CCOA35	CCOA36	CCOA37	CCOA38	CCOA39	CCOA40
Ackley											
MEAN	7.576376E–08	2.075792	2.102422	2.026434	5.696423E–08	3.611941E–10	1.682011E–10	2.031559	1.864850	3.774042E–12[a]	2.920245E–11
SD	2.355940E–07	0.830001	0.848878	0.788147	1.204036E–08	1.330050E–10	2.025819E–10	0.908706	0.793441	1.479200E–12[b]	2.331039E–11
p	1.000000	4.561823E–11	4.561823E–11	4.561823E–11	3.319196E–11	1.925598E–12	1.720251E–12	4.561823E–11	4.561823E–11	1.720251E–12	1.720251E–12
Sphere											
MEAN	6.324382E–13	2.387927	4.206601	2.892663	7.295709E–17	9.662233E–19	2.084946E–20	2.046749	1.052056	8.038961E–24[a]	1.406544E–20
SD	2.765086E–12	4.458472	6.614353	5.153145	4.648786E–17	7.817550E–19	5.958646E–20	4.042535	2.422858	9.072429E–24[b]	5.107536E–20
p	1.000000	4.561823E–11	4.561823E–11	4.561823E–11	1.015245E–11	1.720251E–12	1.720251E–12	4.561823E–11	4.561823E–11	1.720251E–12	1.720251E–12
Shifted sphere											
MEAN	8.934043E–14	2.555065	4.282584	3.065799	5.609401E–17	8.456985E–19	4.350763E–20	2.252082	1.121226	5.605639E–24[a]	8.211737E–21
SD	3.513040E–13	4.793232	7.906116	6.265140	3.512637E–17	6.872715E–19	1.718986E–19	5.141847	3.001584	5.855746E–24[b]	2.477531E–20
p	1.000000	4.561823E–11	4.561823E–11	4.561823E–11	4.211546E–12	1.720251E–12	1.720251E–12	4.561823E–11	4.561823E–11	1.720251E–12	1.720251E–12
Salomon											
MEAN	0.117014	1.345929	1.397313	1.324661	0.194339	0.149873	0.113206	1.330738	1.412376	0.106540[a]	0.109873
SD	0.010630	0.401909	0.415799	0.381867	0.001531	7.712870E–17[b]	1.031897E–17	0.406017	0.393534	1.017213E-7	2.805261E-7
p	1.000000	8.173246E–12	8.173246E–12	8.173246E–12	6.884390E–12	6.884390E–12	7.770594E–13	8.173246E–12	8.173246E–12	7.770594E–13	7.770594E–13
Shifted Rosen-brock											
MEAN	4.514106	1.034545E+04	1.541110E+04	8.508551E+03	11.077304	2.843604	0.984041	6.964152E+03	4.029378E+03	0.762010[a]	2.668739
SD	0.941888	2.429552E+04	3.942060E+04	1.589123E+04	0.418753	0.216953	0.131909	1.526393E+04	8.959652E+03	0.102973[b]	0.260333
p	1.000000	1.406492E–11	4.561823E–11	4.561823E–11	6.551578E–12	1.406492E–11	2.164832E–11	2.983925E–11	4.561823E–11	6.551578E–12	1.132047E–11
Rosenbrock											
MEAN	1.787732	7.531944E+03	1.435259E+04	1.154041E+04	14.247846	2.283266	1.003188	5.895206E+03	4.555008E+03	0.643797[a]	1.011298
SD	0.547281	1.422273E+04	2.955187E+04	2.785058E+04	0.443734	0.202510	0.195952	1.181718E+04	1.108607E+04	0.083505[b]	0.110293

Table 6 (continued)

Function	Algorithms										
	COA	CCOA31	CCOA32	CCOA33	CCOA34	CCOA35	CCOA36	CCOA37	CCOA38	CCOA39	CCOA40
p	1.000000	4.561823E–11	4.561823E–11	4.561823E–11	8.159589E–12	2.681873E–11	8.159589E–12	4.104007E–11	3.691245E–11	8.159589E–12	7.312394E–12
Rastrigin											
MEAN	1.475385E–12	2.552528	3.282978	2.792643	2.503763E–13	0.000000	0.000000	2.429396	2.042322	0.000000[a]	9.473903E–17
SD	5.872915E–12	1.361846	1.397841	1.327197	1.257020E–13	0.000000	0.000000	1.332852	1.181312	0.000000[b]	4.150383E–16
p	NaN	0.333711[c]	0.333711[c]	0.333711[c]	0.333711[c]	NaN	NaN	0.333711[c]	0.333711[c]	NaN	NaN
Penalty 1											
MEAN	2.590853E–15	0.159304	0.214063	0.152950	1.983298E–17	2.118882E–20	2.010576E–21	0.149652	0.118581	8.042310E–25[a]	3.491928E–21
SD	9.673585E–15	0.274027	0.328790	0.251290	1.168706E–17	2.200628E–20	5.288380E–21	0.252164	0.217863	1.013350E–24[b]	1.276640E–20
p	1.000000	4.561823E–11	4.561823E–11	4.561823E–11	3.371781E–12	1.720251E–12	1.720251E–12	4.561823E–11	4.561823E–11	1.720251E–12	1.720251E–12
Penalty 2											
MEAN	6.066838E–14	0.237486	0.278876	0.266261	2.293674E–17	2.286263E–19	1.565736E–20	0.235489	3.309636	5.195125E–24[a]	1.540673E–20
SD	2.605594E–13	0.328719	0.333886	0.338498	1.631336E–17	2.411675E–19	3.872204E–20	0.322126	13.906043	5.140001E–24[b]	4.454932E–20
p	1.000000	4.561823E–11	4.561823E–11	4.561823E–11	1.745784E–11	1.720251E–12	1.720251E–12	4.561823E–11	4.561823E–11	1.720251E–12	1.720251E–12
Schwefel											
MEAN	1.272757E–04	34.170319	35.750476	31.267127	1.272757E–04	1.272757E–04	1.272757E–04	24.398849	16.807200	1.272757E–04[a]	1.272757E–04
SD	1.025796E–10	42.914219	42.862917	39.489621	8.432195E–14	5.013871E–14	1.087239E–13	37.822140	28.008029	0.000000[b]	1.092408E–13
p	1.000000	3.619766E–12	3.619766E–12	3.619766E–12	3.520262E–13	3.520262E–13	3.520262E–13	3.619766E–12	3.619766E–12	3.520262E–13	3.520262E–13

[a] the minimum MEAN value, [b] the minimum SD value, [c] *p* value over 0.05

reveals CCOA24 slightly outperforms others on function 4, 6, 10, but it obviously has the fastest convergence rate on other functions. It's also can be seen clearly in Fig. 6, the performance of each algorithm is visualized by the height of block stacking, according to the data *MEAN* in Table 5. That means algorithms with the Gauss/mouse, Piecewise, Sine, Chebyshev, and Circle chaotic migration operators successfully overtake the basic COA. By and large, the performance of the Gauss/mouse chaotic migration operator is superior to other ten algorithms. This superiority is noteworthy in a statistical way has been proved by *p* value in Table 5.

The data presented in Table 6 has shown that CCOA39 performs better than COA and other algorithms on all of the benchmark functions. In other words, the basic COA combined with the chaotic map named Chebyshev performs best in the comparison. Figure 7 displays the performance level of each algorithm according to the data *MEAN* in Table 6. From Table 6 and Fig. 7, we can know that CCOA39, CCOA40, CCOA36, CCOA35, CCOA34 have better results compared to COA on function 1, 2, 3, 7, 8, 9, 10. And on these functions, the convergence curves drawn in Fig. 8 pass the message that CCOA34, CCOA35, CCOA36, CCOA39, and CCOA40 have better global search ability and faster convergence speed than others. Meanwhile, CCOA31, CCOA32, CCOA33, CCOA37, CCOA38 show worse results in contrast with COA. That is to say that the Chebyshev map improves the performance of COA in terms of updating the social conditions, exploration and exploitation through synthetical consideration.

To summarize, all Tables and Figures show that the employ of chaotic map make the algorithms performed well compared with COA, contains Logistic, Tent, Iterative, Gauss/mouse, Piecewise, Sine, Singer, Sinusoidal, Chebyshev, and Circle. In addition, algorithms in Sects. 3.3, 3.4 are much better than the others in Sects. 3.1, 3.2. The work described in this article demonstrates CCOA has a faster convergence velocity and stronger robustness to the population size in contrast with COA.

4.2 Comparison of CCOA with other algorithms

Chaotic particle swarm optimization (CPSO) 1–5 are proposed in Xiang et al. (2007), Hong-Ji et al. (2004), Liu et al. (2005), Chuanwen and Bompard (2005a, b), Chaotic accelerated PSO (CAPSO) is designed in Gandomi et al. (2013b). In addition, the figure is difficult to draw on account of the over-large reference datum, thereby encouraging y axis range is specified from 0 to 50. The evaluation criterion is that the small mean costs the better algorithm. The results is illustrated in Fig. 9 that the CCOA proposed in this paper

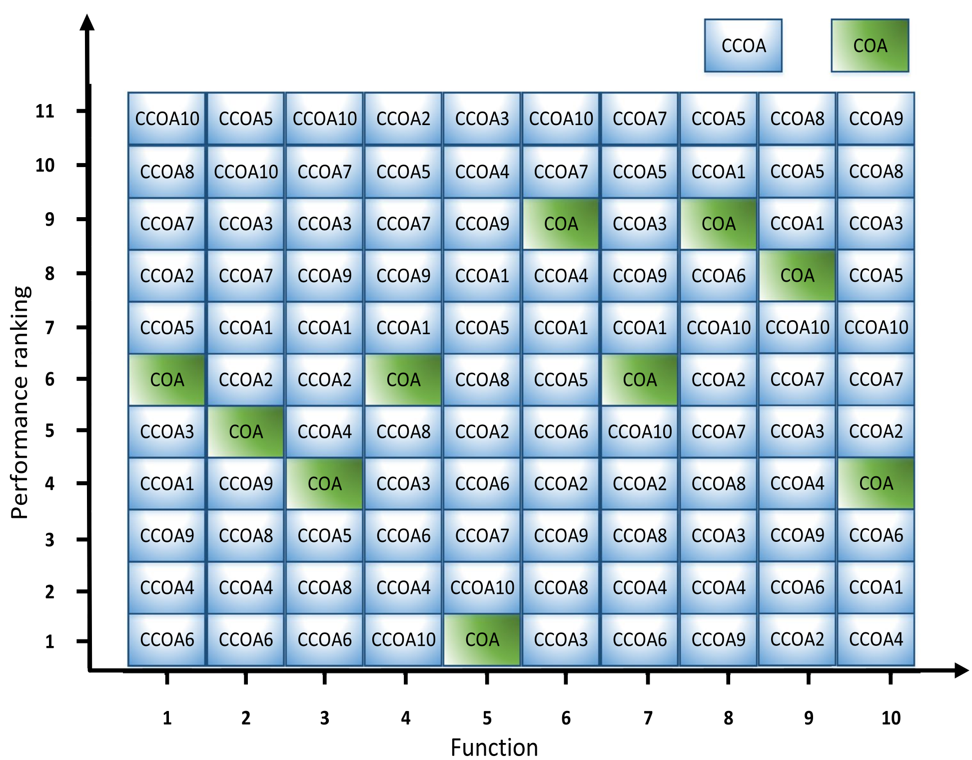


Fig. 1 The performance ranking of convergence speed of CCOA1-10 and COA

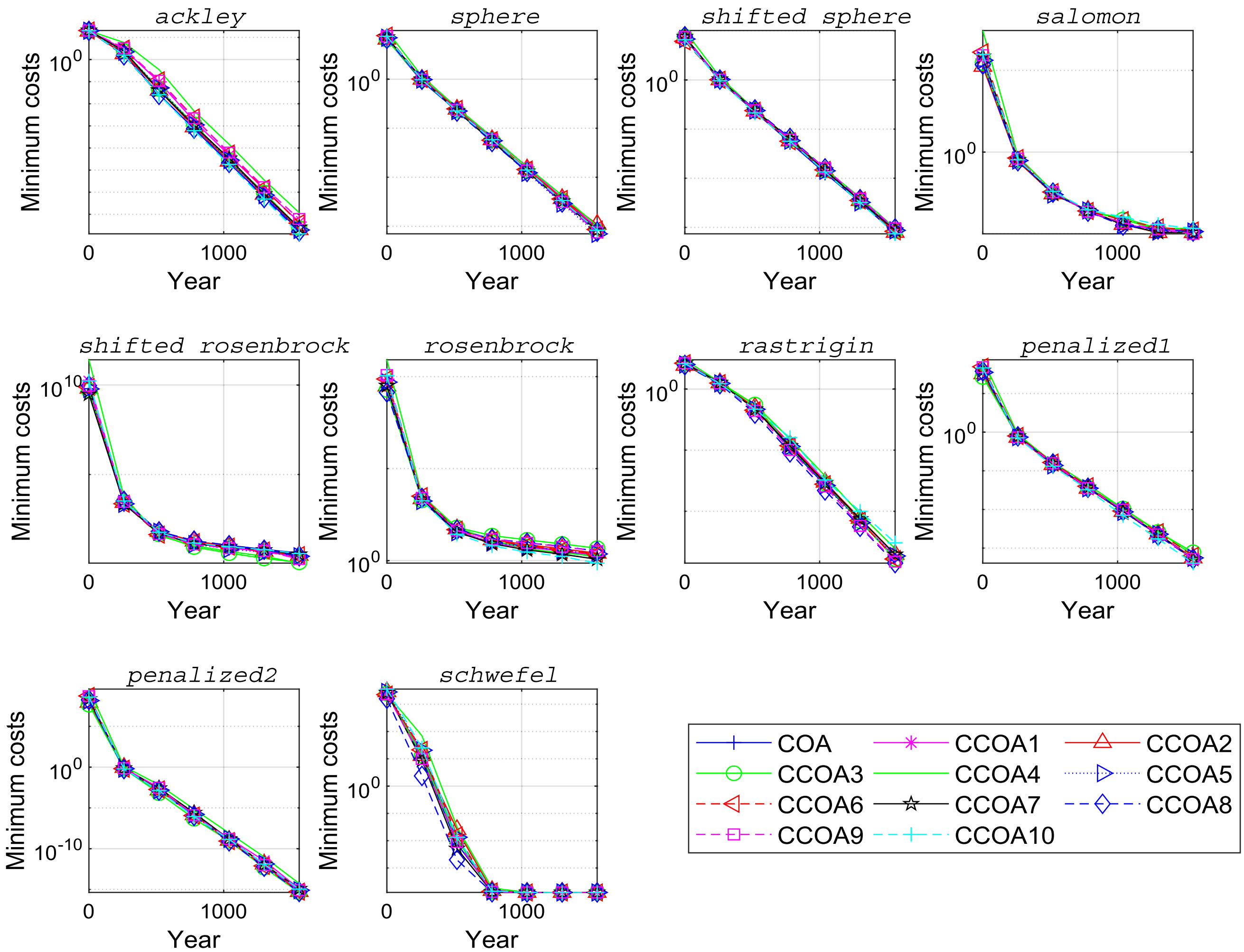


Fig. 2 The result of use of chaotic maps for initializing social condition

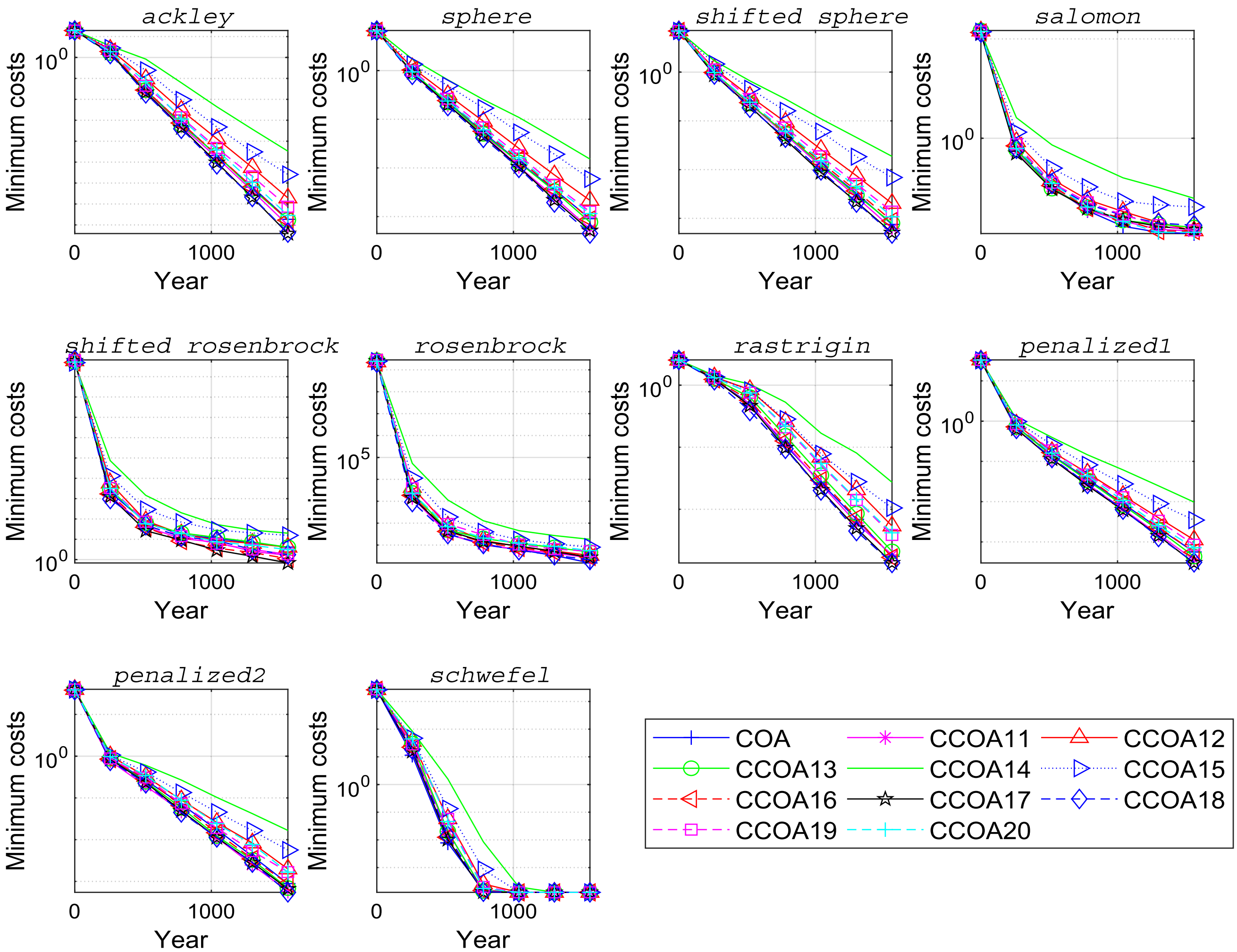


Fig. 3 The use of chaotic maps for updating social condition

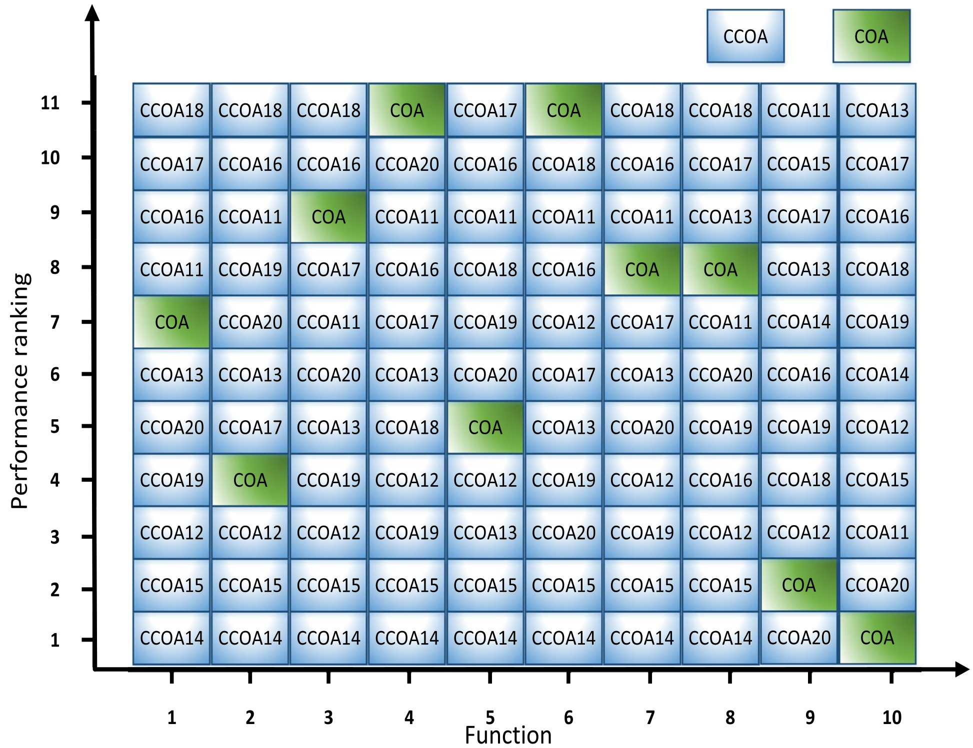


Fig. 4 The performance ranking of convergence speed of CCOA11-20 and COA

has gained better performances with the test by Ackley, Rosenbrock, and Rastrigin functions.

5 Conclusion

An ingenious combinations of chaos theory heighten the performance of the basic COA in this paper. The investigation of ten chaotic maps adjusts some parameters of COA in experiments. In addition, ten benchmark functions are selected to test CCOA's performances in four combination methods. Then the objective comparison reveals CCOA outperforms than other algorithms. For migration, for initializing, updating the social conditions and migration together, the CCOA achieves excellent results. Moreover, the results have also suggested that the Chebyshev map and Gauss/mouse map better than other chaotic maps. For the application in some domains, many algorithms lack the advantages of a simple configuration and as few parameters as possible. However, CCOA differs from them that has the advantages mentioned above.

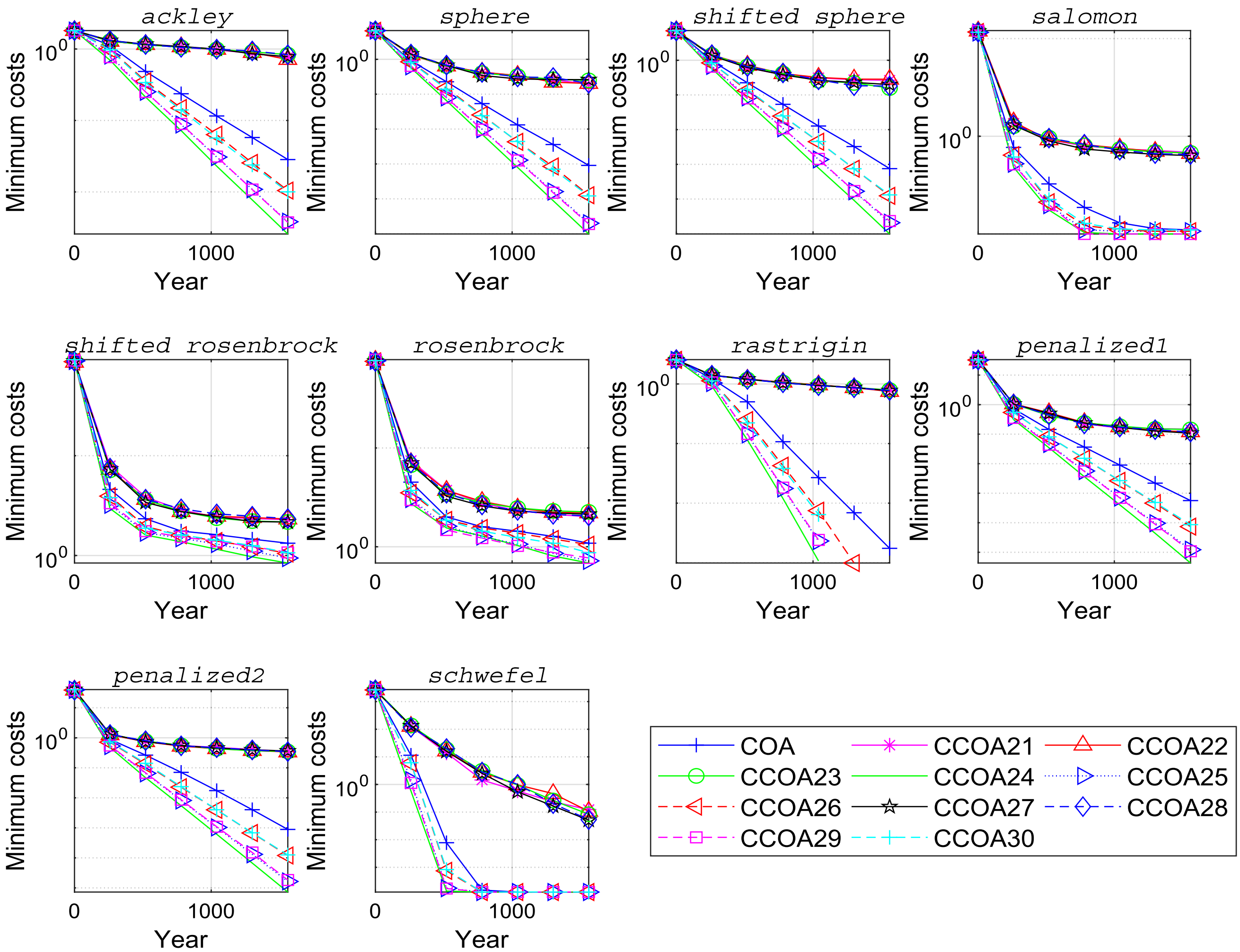


Fig. 5 The use of chaotic maps for the migration of coyotes

Fig. 6 The performance ranking of convergence speed of CCOA21-30 and COA

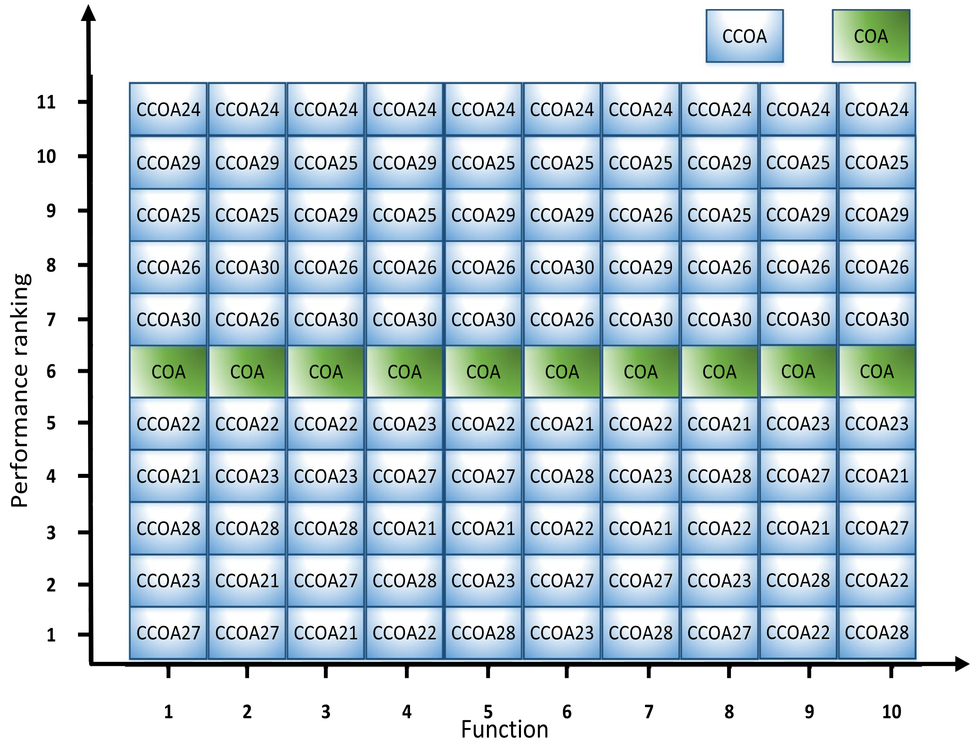
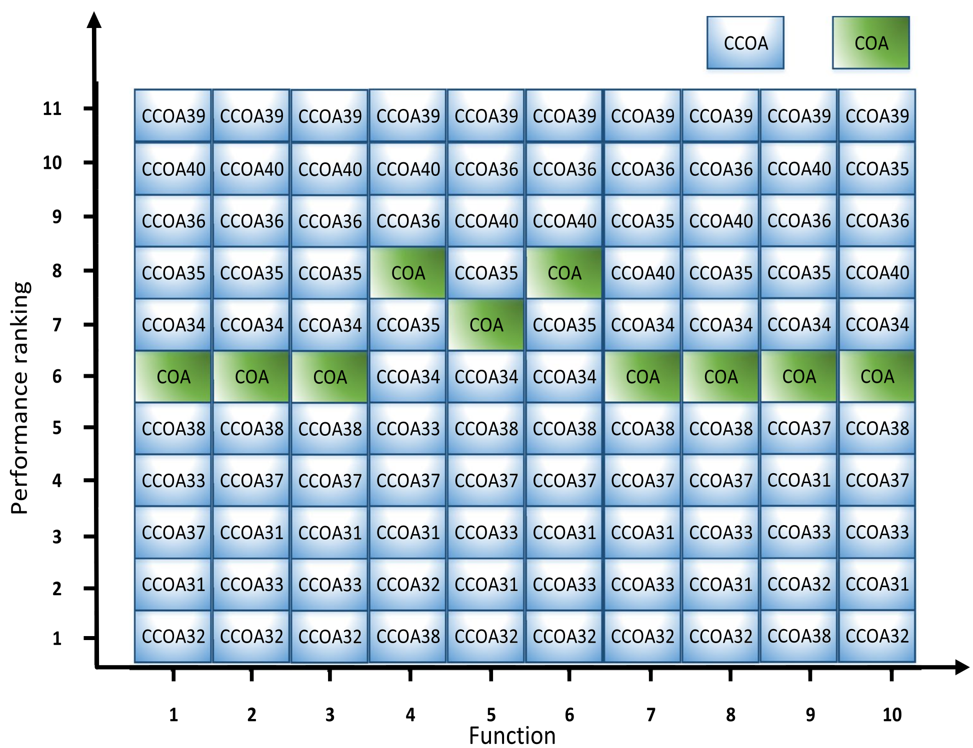


Fig. 7 The performance ranking of convergence speed of CCOA31-40 and COA



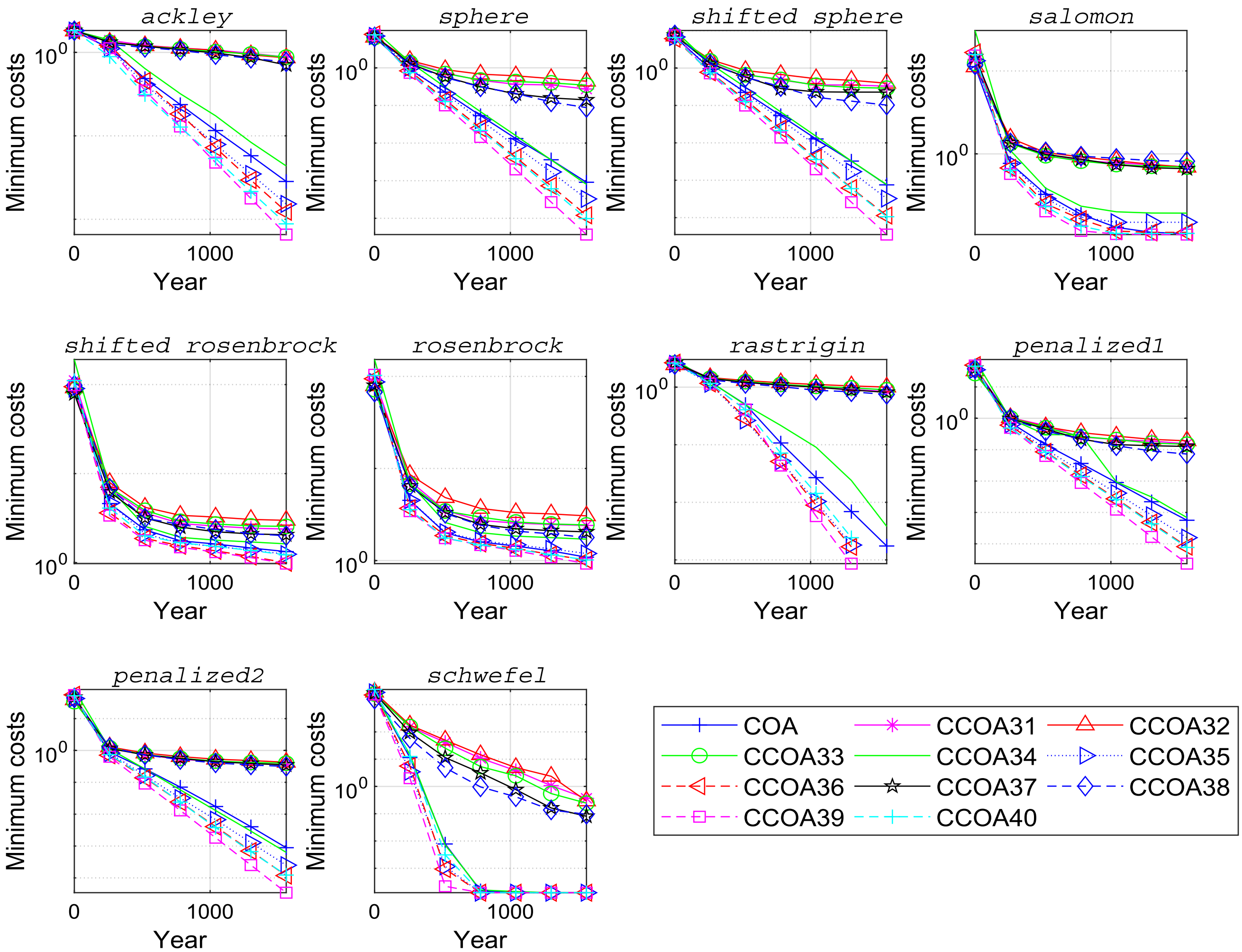
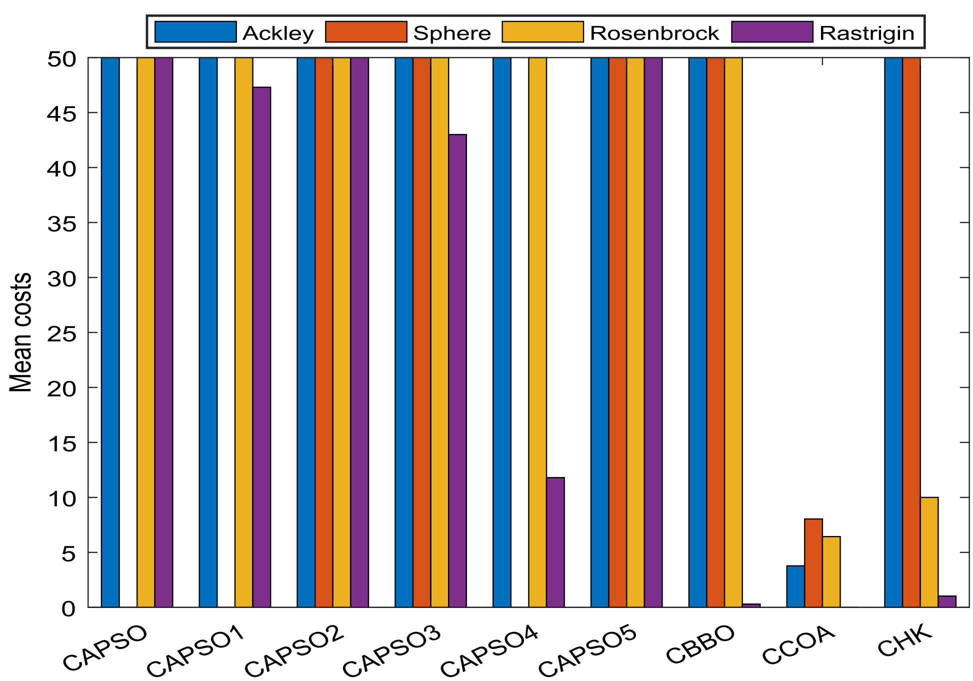


Fig. 8 The use of chaotic maps for initializing social conditions, updating social condition, and the migration of coyotes

Fig. 9 Comparison of CCOA results with other algorithms for Ackley, Sphere, Rosenbrock, Rastrigin functions



Acknowledgements This work was supported by grants from the National Key Research and Development Program of China under Contract No. 2016YFE0200200, the National Natural Science Funds of China under Contract No. 61701253 and 61801240.

References

Agrawal SK, Sahu OP (2015) Artificial bee colony algorithm to design two-channel quadrature mirror filter banks. Swarm Evol Comput 21:24–31. https://doi.org/10.1016/j.swevo.2014.12.001. http://www.sciencedirect.com/science/article/pii/S2210650214000819

Chuanwen J, Bompard E (2005a) A hybrid method of chaotic particle swarm optimization and linear interior for reactive power optimisation. Math Comput Simul 68(1):57–65. https://doi.org/10.1016/j.matcom.2004.10.003. http://www.sciencedirect.com/science/article/pii/S0378475404002599

Chuanwen J, Bompard E (2005b) A self-adaptive chaotic particle swarm algorithm for short term hydroelectric system scheduling in deregulated environment. Energy Convers Manag 46(17):2689–2696. https://doi.org/10.1016/j.enconman.2005.01.002. http://www.sciencedirect.com/science/article/pii/S0196890405000233

Clerc M, Kennedy J (2002) The particle swarm - explosion, stability, and convergence in a multidimensional complex space. IEEE Trans Evol Comput 6(1):58–73. https://doi.org/10.1109/4235.985692

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 Evol Comput 1(1):3–18. https://doi.org/10.1016/j.swevo.2011.02.002. http://www.sciencedirect.com/science/article/pii/S2210650211000034

Dhiman G, Garg M, Nagar A, Kumar V, Dehghani M (2020) A novel algorithm for global optimization: Rat swarm optimizer. J Ambient Intell Humaniz Comput. https://doi.org/10.1007/s12652-020-02580-0

Dorigo M, Caro GD (1999) Ant colony optimization: a new meta-heuristic. In: Proceedings of the 1999 congress on evolutionary computation-CEC99 (Cat. No. 99TH8406), vol 2, pp 1470–1477. https://doi.org/10.1109/CEC.1999.782657

Dorigo M, Birattari M, Stutzle T (2006) Ant colony optimization. IEEE Comput Intell Mag 1(4):28–39

Eberhart R, Kennedy J (1995) A new optimizer using particle swarm theory. In: MHS'95. Proceedings of the sixth international symposium on micro machine and human science, pp 39–43. https://doi.org/10.1109/MHS.1995.494215

Gandomi AH, Yang XS, Talatahari S, Alavi AH (2013a) Firefly algorithm with chaos. Commun Nonlinear Sci Numer Simul 18(1):89–98. https://doi.org/10.1016/j.cnsns.2012.06.009. http://www.sciencedirect.com/science/article/pii/S1007570412002717

Gandomi AH, Yun GJ, Yang XS, Talatahari S (2013b) Chaos-enhanced accelerated particle swarm optimization. Commun Nonlinear Sci Numer Simul 18(2):327–340. https://doi.org/10.1016/j.cnsns.2012.07.017. http://www.sciencedirect.com/science/article/pii/S1007570412003292

Gwo-Ching L, Ta-Peng T (2006) Application of a fuzzy neural network combined with a chaos genetic algorithm and simulated annealing to short-term load forecasting. IEEE Trans Evol Comput 10(3):330–340. https://doi.org/10.1109/TEVC.2005.857075

Hashim FA, Houssein EH, Mabrouk MS, Al-Atabany W, Mirjalili S (2019) Henry gas solubility optimization: a novel physics-based algorithm. Future Gener Comput Syst 101:646–667. https://doi.org/10.1016/j.future.2019.07.015. http://www.sciencedirect.com/science/article/pii/S0167739X19306557

Hong-Ji M, Peng Z, Rong-Yang W, Xiao-Jing H, Zhi X (2004) A hybrid particle swarm algorithm with embedded chaotic search. IEEE Conf Cybern Intell Syst 1:367–371. https://doi.org/10.1109/ICCIS.2004.1460442

Ibrahim RA, Ewees AA, Oliva D, Abd Elaziz M, Lu S (2019) Improved SALP swarm algorithm based on particle swarm optimization for feature selection. J Ambient Intell Humaniz Comput 10(8):3155–3169. https://doi.org/10.1007/s12652-018-1031-9

Jia D, Jiao Y, Zhang J (2009) Satisfactory design of IIR digital filter based on chaotic mutation particle swarm optimization. In: 2009 Third international conference on genetic and evolutionary computing, pp 48–51. https://doi.org/10.1109/WGEC.2009.172

Jiang BLW (1998) Optimizing complex functions by chaos search. Cybern Syst 29(4):409–419. https://doi.org/10.1080/019697298125678

Jinfeng Z (2011) Chaotic particle swarm optimization algorithm based on tent mapping for dynamic origin-destination matrix estimation. In: 2011 International conference on electric information and control engineering, pp 221–224. https://doi.org/10.1109/ICEICE.2011.5777924

Karaboga D, Basturk B (2007) A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. J Glob Optim 39(3):459–471. https://doi.org/10.1007/s10898-007-9149-x. https://doi.org/10.1007/s10898-007-9149-x

Kaur G, Arora S (2018) Chaotic whale optimization algorithm. J Comput Des Eng 5(3):275–284. https://doi.org/10.1016/j.jcde.2017.12.006. http://www.sciencedirect.com/science/article/pii/S228843001730132X

Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceedings of ICNN'95 - international conference on neural networks, vol 4, pp 1942–1948. https://doi.org/10.1109/ICNN.1995.488968

Kohli M, Arora S (2018) Chaotic grey wolf optimization algorithm for constrained optimization problems. J Comput Des Eng 5(4):458–472. https://doi.org/10.1016/j.jcde.2017.02.005. http://www.sciencedirect.com/science/article/pii/S2288430016301142

LüQ Shen G, Yu R (2003) A chaotic approach to maintain the population diversity of genetic algorithm in network training. Comput Biol Chem 27:363–371. https://doi.org/10.1016/S1476-9271(02)00083-X

Liu B, Wang L, Jin YH, Tang F, Huang DX (2005) Improved particle swarm optimization combined with chaos. Chaos Solitons Fractals 25(5):1261–1271. https://doi.org/10.1016/j.chaos.2004.11.095. http://www.sciencedirect.com/science/article/pii/S0960077905000330

Mansouri N, Mohammad Hasani Zade B, Javidi MM (2019) Hybrid task scheduling strategy for cloud computing by modified particle swarm optimization and fuzzy theory. Comput Ind Eng 130:597–633. https://doi.org/10.1016/j.cie.2019.03.006. http://www.sciencedirect.com/science/article/pii/S0360835219301421

Mirjalili S, Lewis A (2016) The whale optimization algorithm. Adv Eng Softw 95:51–67. https://doi.org/10.1016/j.advengsoft.2016.01.008. http://www.sciencedirect.com/science/article/pii/S0965997816300163

Mirjalili S, Mirjalili SM, Hatamlou A (2016) Multi-verse optimizer: a nature-inspired algorithm for global optimization. Neural Comput Appl 27(2):495–513. https://doi.org/10.1007/s00521-015-1870-7. https://doi.org/10.1007/s00521-015-1870-7

Mirjalili S, Gandomi AH, Mirjalili SZ, Saremi S, Faris H, Mirjalili SM (2017) Salp swarm algorithm: a bio-inspired optimizer for engineering design problems. Adv Eng Softw 114:163–191. https://doi.org/10.1016/j.advengsoft.2017.07.002. http://www.sciencedirect.com/science/article/pii/S0965997816307736

Naseri A, Jafari Navimipour N (2019) A new agent-based method for QoS-aware cloud service composition using particle swarm optimization algorithm. J Ambient Intell Humaniz Comput

10(5):1851–1864. https://doi.org/10.1007/s12652-018-0773-8. https://doi.org/10.1007/s12652-018-0773-8

Pierezan J, Coelho LDS (2018) Coyote optimization algorithm: a new metaheuristic for global optimization problems. In: 2018 IEEE congress on evolutionary computation (CEC), pp 1–8. https://doi.org/10.1109/CEC.2018.8477769

Saad E, Elhosseini MA, Haikal AY (2019) Culture-based artificial bee colony with heritage mechanism for optimization of wireless sensors network. Appl Soft Comput 79:59–73. https://doi.org/10.1016/j.asoc.2019.03.040. http://www.sciencedirect.com/science/article/pii/S1568494619301668

Saremi S, Mirjalili S, Lewis A (2014) Biogeography-based optimisation with chaos. Neural Comput Appl 25(5):1077–1097. https://doi.org/10.1007/s00521-014-1597-x. https://doi.org/10.1007/s00521-014-1597-x

Sayed GI, Hassanien AE, Azar AT (2019) Feature selection via a novel chaotic crow search algorithm. Neural Comput Appl 31(1):171–188. https://doi.org/10.1007/s00521-017-2988-6. https://doi.org/10.1007/s00521-017-2988-6

Suganthan P, Hansen N, Liang J, Deb K, Chen YP, Auger A, Tiwari S (2005) Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization. Nat Comput

Voraprateep J (2013) Robustness of Wilcoxon signed-rank test against the assumption of symmetry. Thesis. https://doi.org/10.13140/2.1.3241.3121

Wang GG, Guo L, Gandomi AH, Hao GS, Wang H (2014) Chaotic krill herd algorithm. Inf Sci 274:17–34. https://doi.org/10.1016/j.ins.2014.02.123. http://www.sciencedirect.com/science/article/pii/S0020025514002291

Wang J, Liu C, Wang J, Wu Y, Lin M, Cheng J (2018a) Physical-layer security for indoor visible light communications: secrecy capacity analysis. IEEE Trans Commun 66(12):6423–6436. https://doi.org/10.1109/TCOMM.2018.2859943

Wang J, Zhu J, Lin S, Wang J (2018b) Adaptive spatial modulation based visible light communications: SER analysis and optimization. IEEE Photonics J 10(3):1–14. https://doi.org/10.1109/JPHOT.2018.2834388

Xiang T, Liao X, Kw Wong (2007) An improved particle swarm optimization algorithm combined with piecewise linear chaotic map. Appl Math Comput 190(2):1637–1645. https://doi.org/10.1016/j.amc.2007.02.103. http://www.sciencedirect.com/science/article/pii/S0096300307002081

Yang XS (2010) Firefly algorithms for multimodal optimization, vol 5792. https://doi.org/10.1007/978-3-642-04944-6_14