Coyote Optimization Algorithm with Linear Convergence for Global Numerical Optimization

Hsin-Jui Lin and Sheng-Ta Hsieh(✉)

Asia Eastern University of Science and Technology, New Taipei City, Taiwan
fo013@mail.aeust.edu.tw

Abstract. In particular, the popularity of computational intelligence has accelerated the study of optimization. Coyote Optimization Algorithm (COA) is a new meta heuristic optimization. It is pays attention to the social structure and experience exchange of coyotes. In this paper, the coyote optimization algorithm with linear convergence (COALC) is proposed. In order to explore a huge search space in the pre-optimization stage and to avoid premature convergence, the convergence factor is also involved. Thus, the COALC will explore a huge search space in the early optimization stage to avoid premature convergence. Also, the small area is adopted in the later optimization stage to effectively refine the final solution, while simulating a coyote killed by a hunter in the environment. It can avoid the influence of bad solutions. In experiments, ten IEEE CEC2019 test functions is adopted. The results show that the proposed method has rapid convergence, and a better solution can be obtained in a limited time, so it has advantages compared with other related methods.

Keywords: Functional optimization · Swarm intelligence · Global optimization problems · Coyote Optimization Algorithm · Coyote optimization algorithm with linear convergence

1 Introduction

Optimization can now solve many problems in the real world, including civil engineering, construction, electromechanical, control, financial, health management, etc. There are significant results [1–5], whether it is applied to image recognition, feature extraction, machine learning, and deep learning model training, The optimization algorithm can be used to adjust [6–8]. The optimization method can make the traditional researcher spend a lot of time to establish the expert system to adjust and optimize, and greatly reduce the time required for exploration. With the exploration of intelligence, the complexity of the problem gradually increases.

In the past three decades, meta-heuristic algorithms that simulate the behavior of nature have received a lot of attention, for example, Particle Swarm Optimization (PSO) [9], Differential Evolution (DE) [10], Crow Search algorithm (CSA) [11], Grey Wolf Optimization (GWO) [12], Coyote Optimization Algorithm (COA) [13], Whale Optimization Algorithm (WOA) [14], Honey Badger Algorithm (HBA) [15] and Red fox optimization algorithm (RFO) [16]. These meta-heuristic algorithms inspired by natural behavior are highly efficient in optimization problems. Performance and ease of application have been improved and applied to various problems.

GWO simulates the class system in the predation process of wolves in nature, and divides gray wolves into four levels. Through the domination and leadership between the levels, the gray wolves are driven to find the best solution. This method prevents GWO from falling into the local optimum solution, and use the convergence factor to make the algorithm use a longer moving distance to perform a global search in the early stage, and gradually tends to a local search as time changes. Arora *et al.* [17] mixed GWO and CSA use the flight control parameters and modified linear control parameters in CSA to achieve the balance in the exploration and exploration process, and use it to solve the problem of function optimization.

The idea of COA comes from the coyotes living in North America. Unlike most meta heuristic optimization, which focuses on the predator relationship between predators and prey, COA focuses on the social structure and experience exchange of coyotes. It has a special algorithm structure. Compared with GWO, although alpha wolf (best value) is still used to guide, it does not pay attention to the ruling rules of beta wolf (second best value) and delta wolf (third best value), and balance the global search in the optimization process and local search. Li *et al.* [18] changed the COA differential mutation strategy, designed differential dynamic mutation disturbance strategy and adaptive differential scaling factor, and used it in the fuzzy multi-level image threshold in order to change the COA iteration to the local optimum, which is prone to premature convergence. Get better image segmentation quality.

RFO is a recently proposed meta-heuristic algorithm that imitates the life and hunting methods of red foxes. It simulates red foxes traveling through the forest to find and prey on prey. These two methods correspond to global search and local search, respectively. The hunting relationship between hunters makes RFO converge to an average in the search process.

In this paper, the combination of convergence factors allows COA to implement better exploration, and adds the risk of coyotes being killed by hunters in nature and the ability to produce young coyote to increase local exploration.

In summary, Sect. 2 is a brief review of COA methods and Sect. 3 describes the project scope and objectives of the proposed method. Section 4 shows the experimental results of proposed algorithms and other algorithms on test functions. Finally, the conclusions are in Sect. 5.

2 Standard Coyote Optimization Algorithm

Coyote Optimization Algorithm (COA) was proposed by Pierezan *et al.* [13] in 2018, it has been widely used in many fields due to its unique algorithm structure [19–21]. In COA, the coyote population is divided into N_p groups, and each group contains N_i coyotes, so the total number of coyotes is $N_p * N_i$, at the start of the COA, the number of coyotes in each group has the same population. Each coyote represents the solution of the optimization problem, and is updated and eliminated in the iteration.

COA effectively simulates the social structure of the coyote, that is, the decision variable $\overrightarrow{x}$ of the global optimization problem. Therefore, the social condition *soc* (decision variable set) formed at the time when the i_{th} coyote of the p_{th} ethnic group is an integer t is written as:

$$soc_i^{p,t} = \overrightarrow{x} = \left(x_1, x_2, \ldots, x_d\right) \tag{1}$$

where d is the search space dimension of the optimization problem, the first initialize the coyote race group. Each coyote is randomly generated in the search space. The i_{th} coyote in the race group p is expressed in the j_{th} dimension as:

$$soc_{(i,j)}^{(p,t)} = LB_j + r_j \times (UB_j - LB_j), \quad j = 1,\ 2, \ldots d \tag{2}$$

where LB_j and UB_j represent the lower and upper bounds of the search space, and r_j is a random number generated in the range of 0 to 1. The current social conditions of the i_{th} coyote, as can be shown in (3):

$$fit_i^{p,t} = f\left(soc_i^{p,t}\right) \tag{3}$$

In nature, the size of a coyote group does not remain the same, and individual coyotes sometimes leave or be expelled from the group alone, become a single one or join another group. COA defines the individual coyote outlier probability P_e as:

$$P_e = 0.005 \times N_i^2 \tag{4}$$

When the random number is less than P_e, the wolf will leave one group and enter another group. COA limits the number of coyotes per group to 14. And COA adopts the optimal individual (*alpha*) guidance mechanism:

$$alpha^{p,t} = \left\{soc_i^{p,t} | arg_{i=1,2,\ldots d} \max fit\left(soc_i^{p,t}\right)\right\} \tag{5}$$

In order to communicate with each other among coyotes, the cultural tendency of coyote is defined as the link of all coyotes' social information:

$$cult_j^{p,t} = \begin{cases} O^{p,t}_{\frac{N_i+1}{2},j}, & N_i \textit{ is odd} \\ \dfrac{O^{p,t}_{\frac{N_i+1}{2},j} + O^{p,t}_{\frac{N_i+1}{2},j}}{2}, & \textit{otherwise} \end{cases} \tag{6}$$

The cultural tendency of the wolf pack is defined as the median of the social status of all coyotes in the specific wolf pack, $O^{p,t}$ is the median of all individuals in the population p in the jth dimension at the tth iteration.

The birth of the *pup* is a combination of two parents (coyote selected at random) and environmental influences as:

$$pup_j^{p,t} = \begin{cases} soc^{p,t}_{r_1,j}, & rand_j < P_s \, or \, j = j_1 \\ soc^{p,t}_{r_2,j}, & rand_j \geq P_s + P_a \, or \, j = j_2 \\ R_j, & otherwise \end{cases} \tag{7}$$

Among them, r_1 and r_2 are the random coyotes of two randomly initialized packages, j_1 and j_2 are two random dimensions in the space. Therefore, the newborn coyotes can be inherited through random selection by parents, or new social condition can be produced by random, while P_s and P_a influenced by search space dimension are the scatter probability and the associated probability respectively, as shown in (8) and (9). R_j is a random number in the search space of the j_{th} dimension, and $rand_j$ is a random number between 0 and 1.

$$P_s = 1/d \tag{8}$$

$$P_a = (1 - P_s)/2 \tag{9}$$

In order to maintain the same population size, COA uses coyote group that do not have environmental adaptability ω and the number of coyotes in the same population φ, when $\varphi = 1$ the only coyote in ω dies and $\varphi > 1$ the oldest coyote in ω dies, and pup survives, and when $\varphi < 1$, pup alone cannot satisfy the survival condition. At the same time, in order to show the cultural exchange in the population, set influence led by the alpha wolf (δ_1) and the influence by the group (δ_2), the δ_1 guided by the optimal individual makes the coyote close to the optimal value, and the δ_2 guided by the coyote population reduces the probability of falling into the local optimal value, where cr_1 and cr_2 are Two random coyotes, δ_1 and δ_2 are written as:

$$\delta_1 = alpha^{p,t} - soc_{cr_1}^{p,t} \tag{10}$$

$$\delta_2 = cult^{p,t} - soc_{cr_2}^{p,t} \tag{11}$$

After calculating the two influencing factors δ_1 and δ_2, using the pack influence and the alpha, the new social condition (12) of the coyote is initialized by two random numbers between 0 and 1, and the new social condition (13) (the position of the coyote) is evaluated.

$$new\_soc_i^{p,t} = soc_i^{p,t} + r_1 \times \delta_1 + r_2 \times \delta_2 \tag{12}$$

$$new\_fit_i^{p,t} = f\left(new\_soc_i^{p,t}\right) \tag{13}$$

Finally, according to the greedy algorithm, update the new social condition (the position of the coyote) as (14), and the optimized solution of the problem is the coyote's can best adapt to the social conditions of the environment.

$$soc_i^{p,t+1} = \begin{cases} new\_soc_i^{p,t}, new\_fit_i^{p,t} < fit_i^{p,t} \\ soc_i^{p,t}, otherwise \end{cases} \tag{14}$$

3 Coyote Optimization Algorithm with Linear Convergence

It is important for optimization algorithms to strike a balance between exploration and exploration. In the classic COA, the position update distance (12) of the coyote is calculated by multiplying two random numbers between 0 and 1 and the influencing factors δ_1 and δ_2. This method makes the position of the coyote tend to an average. As a result, the global search capability in the early stage of the algorithm is insufficient, and the local search cannot be performed in depth in the later stage. At the same time, when calculating the social culture of coyotes (6), they will be dragged down by the poorly adapted coyotes, resulting in poor final convergence.

In order to overcome the limitations of the above conventional COA, the linear convergence strategy of GWO is adopted, and the linear control parameter (a) is calculated by follows.

$$a = 2 - (2 \times t/Max_{iter}) \tag{15}$$

And calculate two random moving vectors A to replace two random numbers, so that the algorithm can move significantly in the early stage to obtain a better global exploration, and in the later stage can perform a deep local search, so that the algorithm can converge in a limited time give better results. The value of a is 2 from the beginning of the iteration, and decreases linearly to 0 with the iteration. Therefore, the movement vector A is calculated by follows.

$$A = 2 * \mathrm{a} * r_1 * a \tag{16}$$

Among them, r_1 is a random number between 0 and 1. Therefore, the social conditions of the new coyote will be generated by follows (17). The pseudo code of proposed method is presented (see Fig. 1).

$$new\_soc_i^{p,t} = soc_i^{p,t} + A_1 \times \delta_1 + A_2 \times \delta_2 \tag{17}$$

COA uses the average of coyote information to form social culture, but it is easily affected by the coyote with the lowest adaptability, making iterative early-stage algorithms unable to quickly converge to a better range. Therefore, referring to the hunting relationship between the red fox and the hunter in the RFO, and applying it in the COA to simulate the situation where the coyote strays into the range of human activities and is hunted, the probability of the coyote being killed by the hunter is H (18), by The linear control parameter is calculated and rounded. With time, H will gradually decrease to 0. In the later stage of the algorithm, this mechanism is not used to avoid falling into the local optimal solution.

$$H = [N_i * (a * 0.1)] \tag{18}$$

```
1: Initialize $N_p$ packs with $N_i$ coyotes each (Eq.2).
2: Verify the coyote's adaptation (Eq.3).
3: while stopping criterion is not achieved do
4:     for each p pack do
5:       Define the alpha coyote of the pack (Eq.5).
6:       Compute the social tendency of the pack (Eq.6).
7:       for each c coyotes of the p pack do
8:         Calculate $a$ and $A$(Eq.15 and Eq.16).
9:         Update the social condition (Eq. 17).
10:        Evaluate the new social condition (Eq.13).
11:        Adaptation (Eq.14).
12:      end for
13:      Calculate the $H$(Eq.18).
14:      if $H$ >= 1 then
15:        Replace Coyote(Eq.19).
16:      Birth and death (Eq.7).
17:    end for
18:    Transition between packs (Eq.4).
19:    Update the coyotes' ages.
20: end while
21: Select the best adapted coyote.
```

Fig. 1. Pseudo code of proposed method.

In order to maintain the total population, new coyotes will be produced. Therefore, new coyotes will be born from combining information of the best coyote ($best_1$) and second-best coyote ($best_2$) in the group. The location of the newborn coyotes (19), k is a random vector in the range [0, 1]. Therefore, the pseudocode of COALC is showed below after the formula is replaced.

$$new\_soc_i^{p,t} = k * \frac{soc_{best_1}^{p,t} + soc_{best_2}^{p,t}}{2} \tag{19}$$

4 Experimental Settings and Experimental Results

4.1 Benchmarks Functions and Algorithms Setup

Table 1. Global optimization, dimensions and search range of ten CEC 2019 test functions

No.	Function name	$F_i^* = F_i(x^*)$	D	Range
f_1	Storn's Chebyshev Polynomial Fitting Problem	1	9	[−8192, 8192]
f_2	Inverse Hilbert Matrix	1	16	[−16384, 16384]
f_3	Lennard-Jones Minimum Energy Cluster	1	18	[−4, 4]
f_4	Rastrigin's Function	1	10	[−100, 100]
f_5	Griewank's Function	1	10	[−100, 100]
f_6	Weierstrass Function	1	10	[−100, 100]
f_7	Modified Schwefel's Function	1	10	[−100, 100]
f_8	Expanded Schaffer's F6 Function	1	10	[−100, 100]
f_9	Happy Cat Function	1	10	[−100, 100]
f_{10}	Ackley Function	1	10	[−100, 100]

The proposed COALC method uses the 10 benchmark functions shown in Table 1 in the IEEE CEC2019 test function (CEC2019) [22] to extend our benchmark test, where F_i^* is the global optimum and D is the dimension of the optimization problem. These benchmarks vary according to the number, dimensionality, and search space of the local optimal classifications. In the CEC2019 function, the functions $f_{1,}$ f_2, and f_3 are completely dependent on the parameters and do not rotate (or shift). Among them, f_1 and f_2 are error functions that need to rely on highly conditional solutions, and f_3 is a way to simulate atomic interaction. It is difficult to find the best solution directly in the function f_9, and the optimization algorithm must perform a deep search in the circular groove. F_4, f_5, f_6, f_7, f_8 and f_{10} are classic optimization problems.

In the benchmark test, this paper compared the proposed COALC, standard COA, ICOA, GWO, and RFO. In the experiment, COALC, standard COA and ICOA defined the coyote population number parameter N_p as 6, and the coyote N_i in each group was set as 5. The number of gray wolves and foxes for GWO and RFO is set to 30, and the above optimization algorithms are based on their original settings, only need to mention the parameters of the method. Therefore, all running comparison heuristics have 30 number of species. The experiment was run on Python 3.8.8.

4.2 Comparison and Analysis of Experimental Results

Table 2. Experiment results of five optimizers

No.	COALC	COA [13]	ICOA [18]	GWO [12]	RFO [16]
f_1	0.00E+00 ±4.81E+00	6.42E+05 ±4.77E+05	3.97E+06 ±2.84E+06	8.93E+04 ±1.75E+05	**0.00E+00 ±0.00E+00**
f_2	**3.28E+00 ±5.66E−02**	1.12E+03 ±3.62E+02	2.63E+03 ±9.09E+02	3.99E+02 ±2.83E+02	4.00E+00 ±0.00E+00
f_3	**1.49E+00 ±8.50E−01**	2.27E+00 ±1.46E+00	5.72E+00 ±1.64E+00	1.86E+00 ±1.90E+00	5.47E+00 ±1.90E+00
f_4	**1.26E+01 ±4.26E+00**	1.34E+01 ±5.32E+00	1.78E+01 ±5.89E+00	2.00E+01 ±1.07E+01	9.64E+01 ±1.64E+01
f_5	0.27E+00 ±9.33E−02	**0.18E+00 ±8.31E−02**	0.70E+00 ±1.63E−01	0.86E+00 ±5.45E−01	3.89E+01 ±1.46E+01
f_6	**1.17E+00 ±8.34E−01**	1.92E+00 ±1.26E+00	3.52E+00 ±1.37E+00	1.63E+00 ±1.26E+00	9.03E+00 ±8.84E−01
f_7	5.86E+02 ±2.07E+02	**4.93E+02 ±2.20E+02**	6.39E+02 ±2.30E+02	8.20E+02 ±3.34E+02	2.00E+03 ±2.18E+02
f_8	**2.71E+00 ±4.14E−01**	2.82E+00 ±2.85E−01	3.06E+00 ±3.21E−01	2.82E+00 ±3.87E−01	3.98E+00 ±1.87E−01
f_9	**0.18E+00 ±5.77E−02**	0.22E+00 ±7.12E−02	0.22E+00 ±8.73E−02	0.20E+00 ±7.12E−02	1.37E+00 ±6.52E−01
f_{10}	**1.83E+01 ±5.13E+00**	2.01E+01 ±5.10E−02	2.01E+01 ±5.54E−02	2.05E+01 ±1.20E−01	2.07E+01 ±1.33E−01

Each optimizer is performed 25 independent runs on the CEC2019, and the stopping criterion is equal to the number of ethnic groups * 500 iterations. Thus, the maximum fitness evaluation (FEs) is set as 15,000. The average error obtained from the global optimum and standard deviation is shown in Table 2, and the best performance is shown in bold. In Fig. 2 can be seen that for most of the benchmark functions, COALC can find the best solution compared to other methods. COALC acquires better exploration capabilities by inheriting the relationship between hunter and prey of RFO, and has more outstanding capabilities than other methods in f_1 and f_2 functions.

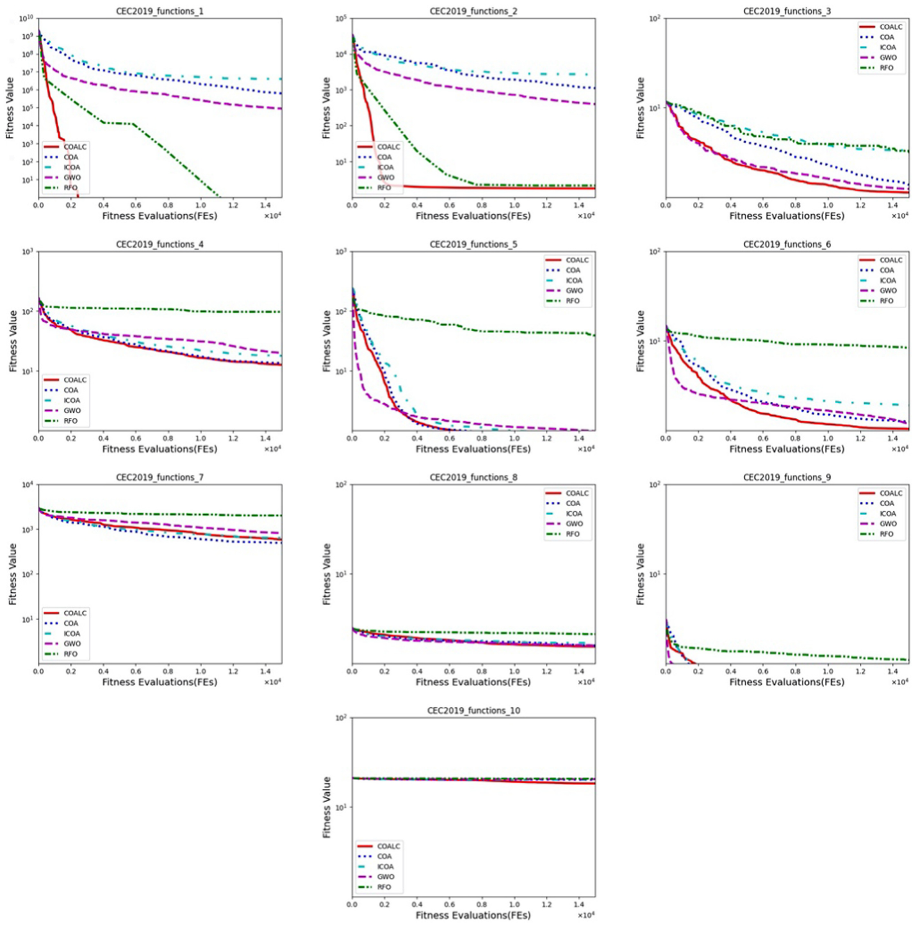

Fig. 2. Median convergence characteristics of five optimizers.

5 Conclusion and Future Research

In this paper, the main contribution is to propose an improved COA algorithm with the convergence factor of the GWO algorithm and the elimination of the worst coyote mechanism, and named it COALC. This method allows COA to acquire better exploration and exploration capabilities in a limited time through the convergence factor, and at the same time eliminates poor coyotes to improve the convergence speed of COA. Finally, Results of experimental benchmark tests have shown that the proposed COALC and recent metaheuristic algorithms such as COA, ICOA, GWO, and RFO, etc., and are evaluated in the CEC2019 test function. In most cases, a better global solution can be obtained than other algorithms.

References

1. Cheng, M., Tran, D.: Two-phase differential evolution for the multiobjective optimization of time–cost tradeoffs in resource-constrained construction projects. IEEE Trans. Eng. Manage. **61**(3), 450–461 (2014)
2. Al-Timimy, A., et al.: Design and losses analysis of a high power density machine for flooded pump applications. IEEE Trans. Ind. Appl. **54**(4), 3260–3270 (2018)
3. Chabane, Y., Ladjici, A.: Differential evolution for optimal tuning of power system stabilizers to improve power systems small signal stability. In: Proceedings of 2016 5th International Conference on Systems and Control (ICSC), pp. 84–89 (2016)
4. Münsing, E., Mather, J., Moura, S.: Blockchains for decentralized optimization of energy resources in microgrid networks. In: Proceedings of 2017 IEEE Conference on Control Technology and Applications (CCTA), pp. 2164–2171 (2017)
5. Lucidi, S., Maurici, M., Paulon, L., Rinaldi, F., Roma, M.: A simulation-based multiobjective optimization approach for health care service management. IEEE Trans. Autom. Sci. Eng. **13**(4), 1480–1491 (2016)
6. Huang, C., He, Z., Cao, G., Cao, W.: Task-driven progressive part localization for fine-grained object recognition. IEEE Trans. Multimed. **18**(12), 2372–2383 (2016)
7. Mistry, K., Zhang, L., Neoh, S.C., Lim, C.P., Fielding, B.: A micro-GA embedded PSO feature selection approach to intelligent facial emotion recognition. IEEE Trans. Cybern. **47**(6), 1496–1509 (2017)
8. Haque, M.N., Noman, M.N., Berretta, R., Moscato, P.: Optimising weights for heterogeneous ensemble of classifiers with differential evolution. In: Proceedings of 2016 IEEE Congress on Evolutionary Computation (CEC), pp. 233–240 (2016)
9. Eberhart, R., Kennedy, J.: A new optimizer using particle swarm theory. In: Proceedings of Sixth International Symposium on Micro Machine and Human Science (MHS), pp. 39–43 (1995)
10. Storn, R., Price, K.: Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. J. Glob. Optim. **11**(4), 341–359 (1997)
11. Askarzadeh, A.: A novel metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm. Comput. Struct. **169**, 1–12 (2016)
12. Mirjalili, S., Mirjalili, S.M., Lewis, A.: Grey wolf optimizer. Adv. Eng. Softw. **69**, 46–61 (2014)
13. Pierezan, J., Coelho, L.D.S.: Coyote optimization algorithm: a new metaheuristic for global optimization problems. In: Proceedings of 2018 IEEE Congress on Evolutionary Computation (CEC), pp. 1–8 (2018)
14. Mirjalili, S., Lewis, A.: The whale optimization algorithm. Adv. Eng. Softw. **95**, 51–67 (2016)
15. Hashim, F.A., Houssein, E.H., Hussain, K., Mabrouk, M.S., Al-Atabany, W.: Honey Badger Algorithm: new metaheuristic algorithm for solving optimization problems. Math. Comput. Simul. **192**, 84–110 (2021)
16. Połap, D., Woźniak, M.: Red fox optimization algorithm. Expert Syst. Appl. **166** ,114107 (2021)
17. Arora, S., Singh, H., Sharma, M., Sharma, S., Anand, P.: A new hybrid algorithm based on grey wolf optimization and crow search algorithm for unconstrained function optimization and feature selection. IEEE Access **7**, 26343–26361 (2019)
18. Li, L., Sun, L., Xue, Y., Li, S., Huang, X., Mansour, R.F.: Fuzzy multilevel image thresholding based on improved coyote optimization algorithm. IEEE Access **9**, 33595–33607 (2021)
19. Abdelwanis, M.I., Abaza, A., El-Sehiemy, R.A., Ibrahim, M.N., Rezk, H.: Parameter estimation of electric power transformers using coyote optimization algorithm with experimental verification. IEEE Access **8**, 50036–50044 (2020)

20. Diab, A.A.Z., Sultan, H.M., Do, T.D., Kamel, O.M., Mossa, M.A.: Coyote optimization algorithm for parameters estimation of various models of solar cells and PV modules. IEEE Access **8**, 111102–111140 (2020)
21. Boursianis, A.D., et al.: Multiband patch antenna design using nature-inspired optimization method. IEEE Open J. Antennas Propag. **2**, 151–162 (2021)
22. Price, K.V., Awad, N.H., Ali, M.Z., Suganthan, P.N.: Problem definitions and evaluation criteria for the 100-digit challenge special session and competition on single objective numerical optimization. Technical Report. Nanyang Technological University, Singapore (2018)