Received: 12 May 2023 | *Revised: 20 June 2023* | *Accepted: 28 June 2023*
DOI: 10.37917/ijeee.20.2.16 *Vol. 20 | Issue 2 | December 2024*

Open Access

Iraqi Journal for Electrical and Electronic Engineering
Original Article

A New Algorithm Based on Pitting Corrosion for Engineering Design Optimization Problems

Hussien A. Al-mtory *[1], Falih M. Alnahwi[2], Ramzy S. Ali[2]
[1]Department of inspection, Iraqi drilling company, ministry of oil, Iraq
[2]Department of Electrical Engineering, College of Engineering, University of Basrah, Basrah, 61001 Iraq

Correspondance
*Hussien A. ALmtory
Department of Electrical Engineering
College of Engineering, University of Basrah, Iraq
Email: pgs.hussien.abduldaym@uobasrah.edu.iq

Abstract
This paper presents a new optimization algorithm called corrosion diffusion optimization algorithm (CDOA). The proposed algorithm is based on the diffusion behavior of the pitting corrosion on the metal surface. CDOA utilizes the oxidation and reduction electrochemical reductions as well as the mathematical model of Gibbs free energy in its searching for the optimal solution of a certain problem. Unlike other algorithms, CDOA has the advantage of dispensing any parameter that need to be set for improving the convergence toward the optimal solution. The superiority of the proposed algorithm over the others is highlighted by applying them on some unimodal and multimodal benchmark functions. The results show that CDOA has better performance than the other algorithms in solving the unimodal equations regardless the dimension of the variable. On the other hand, CDOA provides the best multimodal optimization solution for dimensions less than or equal to (5, 10, 15, up to 20) but it fails in solving this type of equations for variable dimensions larger than 20. Moreover, the algorithm is also applied on two engineering application problems, namely the PID controller and the cantilever beam to accentuate its high performance in solving the engineering problems. The proposed algorithm results in minimized values for the settling time, rise time, and overshoot for the PID controller. Where the rise time, settling time, and maximum overshoot are reduced in the second order system to 0.0099, 0.0175 and 0.005 sec., in the fourth order system to 0.0129, 0.0129 and 0 sec, in the fifth order system to 0.2339, 0.7756 and 0, in the fourth system which contains time delays to 1.5683, 2.7102 and 1.80 E-4 sec., and in the simple mass-damper system to 0.403, 0.628 and 0 sec., respectively.
In addition, it provides the best fitness function for the cantilever beam problem compared with some other well-known algorithms.

Keywords
Corrosion Diffusion, Pitting Corrosion, Energy Level, Benchmark Function, PID Controller.

I. INTRODUCTION

Nature is the main inspiration source of many scientific laws and theories. Therefore, natural phenomena that occur automatically without intervention are a source that often attracts the attention of scientists and researchers. The natural behaviors are usually used for the purpose of deriving equations, laws, and working principles for many industrial and engineering applications. The behavior of various kinds and types of creatures can be utilized in different ways for simulating certain industrial or scientific process. For example, many laws and algorithms were inspired from some behaviors of living organisms [1] such as organs of the living systems, the collective way of life of these living beings, their way of living, or their mass migration. Most of the algorithms are inspired from some natural behaviors like bio algorithms, physics, chemical reactions, and changes of elements in nature. Many algorithms have been invented to emulate the natural systems in such a way that serves the industrial and scientific kinds of problems. The following are the main examples of these algorithms. ant colony (AC) algorithm [2] was proposed in 1992 as an algorithm that follows the ant colony intelligence of communication in solving the optimization problems. In this algorithm, the searching efficiency was improved by increasing the randomness to present variety in the solutions and avoiding falling in local optima [3]. In 1995, particle swarm optimization (PSO) [4] was proposed. This algorithm is based on migration movement of a swarm of birds over a wide searching area. In 2010, an algorithm called cuckoo search algorithm (CS) [5] that relies on the cuckoo species co-parsing in parasitism was presented. An algorithm was proposed in 2012 known as flower pollination algorithm (FPA) [6]. This algorithm is based on the different pollination methods used in agricultural fields and orchards. In addition, there are some algorithms that were inspired from the natural behavior of elements via their chemical reactions without the presence of catalyst such as Multi objective Atomic Orbital Search (MOAOS) [7]. Furthermore, some algorithms were created based on the physical changes such as big bang-big crunch algorithm [8], central force optimization [9] and An improved pedestrian dead reckoning algorithm based on smartphone [10].

Generally, there are many classifications for the optimization algorithms. However, in this work, the classification that is relied on follows the source of inspiration as in [1] at which the algorithms are divided into four categories:

1) Swarm intelligence (SI) based
2) Bio-inspired, but not SI based
3) Physics and chemistry based
4) Other algorithms

The SI algorithms are the most popular type of algorithms. All SI algorithms use multi–agent or multi-particle. There are many reasons for this popularity. One of the most important reasons is the information- sharing between particles or agents which results in faster convergence to the optimum situation. Another reason for the popularity of SI algorithms is that it can be parallelized easily, so it can easily be worked in large scale of optimization. There are many examples of this type algorithms such as Modified camel travelling behavior algorithm (MCA) [11], Artificial bee colony (ABC) [12], Bat algorithm (BA) [13], Bee colony optimization(BCO) [14], and Virtual ant algorithm (VA) [15]. The second type of this classification is the bio-inspired that are not SI-based algorithms. These algorithms depend on the behavior of a certain organism or creature regardless its communication with other mates. Fungi kingdom expansion algorithm [16], biogeography-based optimization [17], brain storm optimization [18], fish-school search [19], and shuffled frog leaping algorithm [20] are examples of this type of algorithms. The third inspiration comes from the physical and chemical properties. The physical algorithms are inspired from the physical theories and laws, while the chemical algorithms are inspired from the interactions and the chemical properties of materials. There are many algorithms that are physically and chemically inspired such as black hole [21], charge system search [22], and water cycle algorithm [23]. The algorithm introduces in this paper which is called Corrosion Diffusion Optimization Algorithm (CDOA) is categorized in this type of classification. Finally, the last type of this classification is called other algorithms. Sometimes, the researchers introduce and develop algorithms that are difficult to include under the above three types. It may be an algorithm inspired from nature but it is not bio, physically, or chemically. Therefore, it is placed under the above mentioned category. Examples of this type are anarchic society optimization [24] and artificial cooperative search [25]. Variety of optimization algorithms are proposed due to several reasons such as the diversity and multiplicity of problem in engineer and industrial applications, and there is no algorithm that solves all engineering and industrial applications problems according to No Free Lunch (NFL) theorem [26].

In this paper, a new algorithm called Corrosion Diffusion Optimization Algorithm (CDOA) is presented. It is based on the corrosion distribution on the metal surfaces. This paper is organized as follows. Section II discusses the method from which the algorithm is inspired. The implementation of the proposed algorithm is demonstrated in Section III, while Section IV verifies the performance of the proposed algorithm with the aid of some benchmark functions to compare it with some other well-known algorithms. Some engineering applications are presented in Section V to fortify the superiority of this algorithm. Section VI is the final conclusion of this research.

II. BEHAVIOR OF CORROSION DIFFUSION

This section demonstrates the inspiration method that was followed in presenting the proposed algorithm. Therefore, this section is divided into the following two subsections.

A. *Inspiration Method*

Corrosion can be defined as a failure or dissolution of a metal or alloy due to a chemical or electrochemical interaction with the medium surrounding it. When the corrosion is caused by physical reasons, it is called wear. However, the corrosion that results from chemical factors and with the help of mechanical factors takes other names such as erosion corrosion and retting corrosion[27], respectively. The study of corrosion is an important matter for humanity because of its impact on the everyday life. The corrosion may results in a huge financial

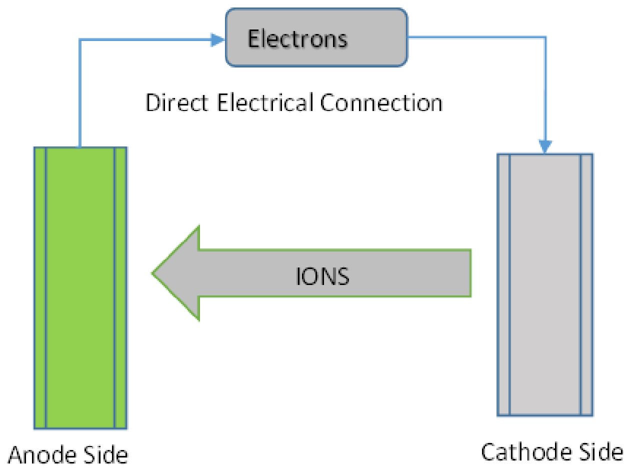

Fig. 1. Structure of the corrosion cell.

wastage, depletion in nature resources, and human discomfort [28]. In general, the dry corrosion and wet corrosion are the main two types of corrosion. The dry corrosion occurs when the metal surface is exposed to dry hot gas. The wet corrosion refers to the exposure of the metal surface to aqueous solution acid or alkali which represents an electrochemical reaction [29]. The chemical corrosion results from a direct reaction between a metal and its surrounding environment or another metal without any catalyst. An example of this corrosion is the corrosion of the reaction of iron with acid chloride hydrogen to generate the corrosion of the iron and liberation of hydrogen. On the other hand, the electrochemical corrosion takes place through an electrochemical interaction between the metal and the surrounding environment. In fact, the electrochemical corrosion depends on oxidation and reduction. The oxidation is the metal's loss of electrons, whereas the second interaction is gaining an electron from the surrounding environment. The two reactions can also be described by the following two chemical equations [30],
where M represents an arbitrary metal and the surrounding environment is represented by X.

$$M \rightarrow Mn^{+} + ne \tag{1}$$

Equation 1 represented Oxidation reaction

$$Nx^{+} + ne^{-} \rightarrow Xn \tag{2}$$

Equation 2 represented Reduction reaction
The corrosion can be expressed by a corrosion cell whose components are illustrated in Fig. 1. The corrosion does not occur if any of the corrosion cell components is missing [31],[32]. These components are as follows: 1) Anode: represents the metal or the site in which the oxidation reaction (loss of electron) occurs.
2) Cathode: represents the metal or the site in which the reaction of reduction (gaining electron) occurs.
3) Electrolyte: is conductive medium between the anode and cathode.
4) Electrical connection between anode and cathode.
5) Potential difference should be available between the anode and cathode to move the electrons from anode to cathode.
The conditions that may lead to an increase in the corrosion rate and may lead to a decrease in the corrosion rate are as follows [33]:
- Effect of oxygen and oxidizer: The corrosion rate increases with the increase of oxygen.
- Effect of velocity: the velocity has the same effect as the oxygen since the velocity controls the polarity of corrosion.
- Effect of temperature: since corrosion is a chemical reaction, the rate of corrosion increases exponentially when the temperature increases.
- Effect of corrosive concentration.
- Effect of galvanic coupling [34].

B. Pitting Corrosion

The corrosion can take different forms depending on the emergence and the environmental conditions that led to its occurrence [35]. Some of the most common forms of corrosion are uniform corrosion, intergranular corrosion, galvanic corrosion, selective corrosion, crevice corrosion, erosion corrosion, stress corrosion, and pitting corrosion. Our proposed algorithm is inspired from the pitting corrosion.The pitting corrosion has the mechanism of the phenomenon of the electrochemical corrosion that is explained previously in this section. A break occurs in the area of a passive layer so that a small part of the metal surface is exposed to the external environment. If the environment conditions are suitable to produce a potential difference to drive the current between the metal and the surrounding environment, the electrochemical reaction does exist. In other words, the base metal is considered as the anode, and the surrounding environment is considered as the cathode of electrochemical cell. Therefore, the electron moves from the metal to the environment [36].
The pitting corrosion passes through three steps. These steps are the initiation or nucleation of pits, pit growth or pit propagation process, and re-passivation of pits [33]. The stability of pit growth depends on the electrolyte, type of metals, and pit-bottom potential. Finally, the spontaneous occurrence and absence of the corrosion can be determined with the aid of the Gibbs free energy [33]. This energy is mathematically expressed in the following formula [31]:

$$\Delta G = -nFE_{cell} \tag{3}$$

where ΔG is the Gibb's free energy, n represents electron transfer from anode side to cathode side, F is faraday constant, and

E_cell denotes the electrical cell potential(E_cathode+E_anode). When $\Delta G < 0$, corrosion becomes spontaneous, while $\Delta G > 0$ results in nonspontaneous corrosion[33].

III. Implementation of Corrosion Diffusion Optimization Algorithm (CDOA)

The proposed algorithm in this study is mainly based on the pitting corrosion diffusion. As mentioned earlier, this type of corrosion goes through three stages. Firstly, it passes though the initiation stage, where the onset of this type of corrosion is random on the surface of the metal. Therefore, suppose that there are a number of pits on the surface of the metal which represents the number of pitting (W). Each of these pits has a number of electrons and ions that move from the positive side which is represented by the surface of the metal to negative side which is represented by the surrounding environment. The number of transferred electrons or ions is symbolized by (N). Actually, N is the dimension of each variable x that represents each pit. Therefore, it is possible to describe the case of the x at the iteration (iter) as in the below matrix:

$$x^{\text{iter}} = \begin{pmatrix} x_{1,1^{iter}} & \cdots & x_{1,N^{iter}} \\ \vdots & \ddots & \vdots \\ x_{W,1^{iter}} & \cdots & x_{W,N^{iter}} \end{pmatrix} \tag{4}$$

where $N = (1, 2,\ldots, D)$, the D represents the dimension of each variable, $W = (1,2,\ldots,$number of pitting$)$, and the iteration number $iter = (1,2,\ldots, iter_{max})$.
Initially, the pitting distributes randomly on the surface of the metal as described in the following formula:

$$x_n^{i.iter} = x_{min} + RAND * (x_{max} - x_{min}) \tag{5}$$

where $RAND$ is random number uniformly distributed between 0 and 1, and x_{max} , x_{min} represent the boundary reign on the metal where x_{max} is the maximum limited of transfer electrons or ions, and x_{minis} the minimum limited of transfer electron or ions. It has already been clarified that the environmental conditions surrounding the corroded metal have a very significant impact on the corrosion rate. It is possible to increase or decrease the corrosion rate by changing the environmental conditions. For this reason, equation (6) combines the effect of the conditions mentioned in Section II.A in the following form:

$$ENV_{EFFECT} = 1 + \frac{ENV_{FACTOR} - LO_{EN_{CON}}}{HI_{EN_{CON}} - LO_{EN_{CON}}} \tag{6}$$

where ENV_{EFFECT} represents the environmental effects on corrosion rate, $HI_{EN_{CON}}$ is the high environmental condition, $LO_{EN_{CON}}$ is the low environmental condition, and ENV_{Factor}denotes the random disruption between the high environmental condition and low environmental condition given by:

$$ENV_{factor} = LO_{EN_{CON}} + RAND(HI_{EN_{CON}} - LO_{EN_{CON}}) \tag{7}$$

The previous subsection also reveals that the stability and growth of the pit mainly depend on the amount of electrical potential difference between the electrodes of the corrosion cell as well as Gibbs free energy equation. For the purpose of determining the stability of the pit, it is necessary to have a decision equation which depends on the potential difference of cell and the number of transferred electrons as given in (8):

$$\begin{aligned} energylevel = {} & -N * uniform\_rand[low_{voltage}, high_{voltage}] \\ & * (x_{oldbest} - x_{privous}) \end{aligned} \tag{8}$$

where $(energylevel)$ represents Gibb's free energy, (uniform rand) represents a uniform disruption between low and high stander level voltages, $x_{oldbest}$ is the global best value, and $x_{privous}$ is the previous iteration value of x.
At each iteration, the sum of the row, which represents the sum of the energy of each pit, is named (sum of energy level). The sum of energy that represents Gibb's free energy is used to determine the stability of the pit as given below two conditions:

CASE ONE : THE UNSTABLE STATE

In this case, the sum of energy level is less than zero, and the potential difference exists. Therefore, it is considered as an unstable situation, and by sure it is not the optimum solution. The updating equation is as follows:

$$x_d^{i\ iter} = x_d^{i\ iter-1} + ENV_{EFFECT} * (x_d^{i\ iter\ best} - x_d^{i\ iter-1\ best}) \tag{9}$$

CASE TWO : THE STABLE STATE

This case happens when the sum of energy level is greater than or equal to zero, or when the potential difference is equal to zero. Therefore, it is considered a stable state in which the value of x remains as it is in the previous iteration.

$$x_d^{i\ iter} = x_d^{i\ iter-1} \tag{10}$$

Fig. 2. and Alg. 1. show the flow chart and the pseudo-code of the proposed algorithm, respectively.

IV. Performance Comparison Based on Benchmark Functions

To verify the performance and efficiency of the proposed algorithm, different types of benchmark functions were used.

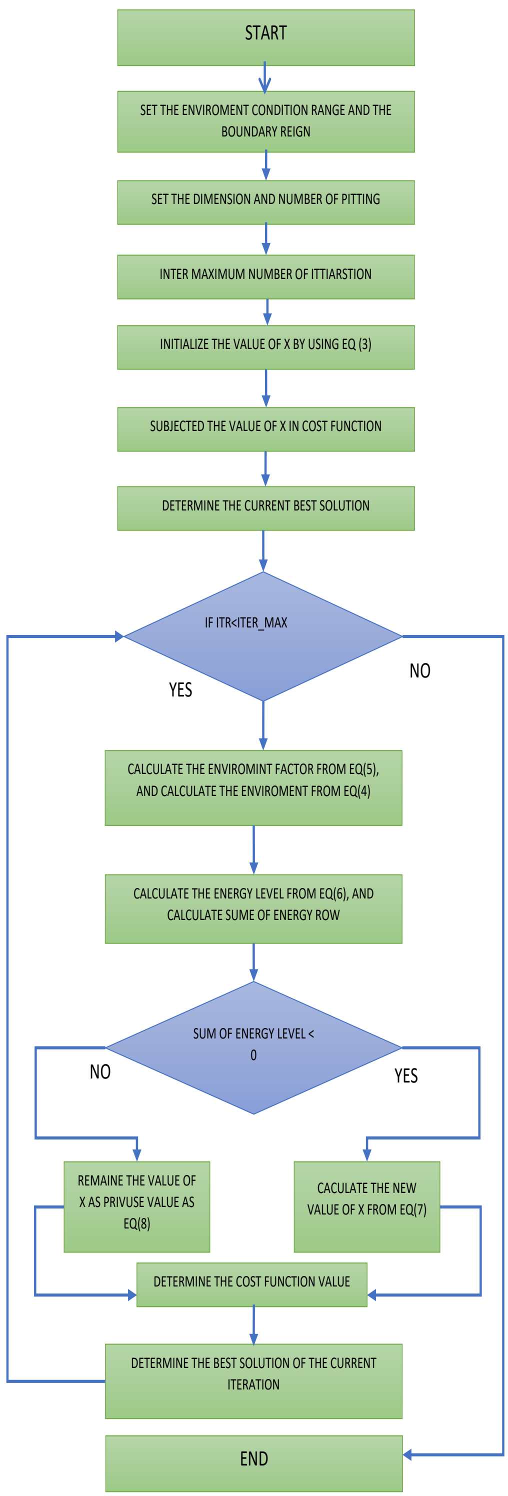

Fig. 2. Flow chart for corrosion diffusion optimization algorithm (CDOA).

CDOA was applied to two sets of benchmark functions [37], namely unimodal and multimodal functions as listed in Table I. The algorithm has been performed using MATLAB R2020a. The proposed algorithm is compared with the following algorithms to highlight its high performance:
A. particle swarm optimization [4].
B. A modified camel algorithm [11].
C. Crow Searching Algorithm [38].
D. Sine Cosine Optimization [39].

Algorithm 1 Corrosion Diffusion Optimization Algorithm (CDOA)

1. Set the range of environment condition, cathode voltage, and anode voltage.
2. Set number of pitting (w) and number of electrons transferred (N) and number of maximum iteration.
3.Initialized the cell voltage from
$$x_n^{i.iter} = x_{min} + RAND * (x_{max} - x_{min})$$
4.Determine the value of cost function.
5.Determine the current best voltage.
6.For $I < itramax$.
7.Initialized environment factor from
$$ENV_{factor} = LO_{EN_{CON}} + RAND(HI_{EN_{CON}} - LO_{EN_{CON}})$$
8. Determine the environment effect from
$$ENV_{EFFECT} = 1 + \frac{ENV_{FACTOR} - LO_{EN_{CON}}}{HI_{EN_{CON}} - LO_{EN_{CON}}}$$
9. Calculate the energy level in
$$uniform\_rand[low_{voltage}, high_{voltage}] * (x_{oldbest} - x_{privous})$$
10.Determine the sum of the row.
11. If sum of row < 0
12. Calculate the update of x as
$$x_d^{i\ iter} = x_d^{i\ iter-1} + ENV_{EFFECT} * (x_d^{i\ iter\ best} - x_d^{i\ iter-1\ best})$$
13. Else .
14. Remain the value of x as pervious value
$$x_d^{i\ iter} = x_d^{i\ iter-1}$$
15.End if.
16.Subject the resulted pitting to the fitness function
17.Determine the current best voltage
18.End for.
19.Determine the best voltage

E. Fungi kingdom algorithm [16].
Different variable dimensions (5, 10, 15, and 20) were applied on each algorithm. The comparison has been carry out under the same conditions for all algorithms, where the maximum iteration is equal to 1000, the variable size is equal to 50, and the dimensions of each variable are as mentions above. Note that all benchmark functions have a minimum when the variable value is equal to zero. For the purpose of comparison

TABLE I.
SOME BENCHMARK FUNCTIONS

#FUN	Name	Range	Function
FUN1	De Jong's sphere function	[–5.12, 5.12]	Unimodal
FUN2	Hyper-ellipsoid function	[–5.12, 5.12]	Unimodal
FUN3	Sum of different power	[–1, 1]	Unimodal
FUN4	Ackley's function	[–32.768, 32.768]	Multimodal
FUN5	Griewangk's function	[–600, 600]	Multimodal
FUN6	Rastrigin's function	[–5.12, 5.12]	Multimodal

between CDOA and the above algorithms, the acceptable error percentage should be determined. Equation (11) shows the error percentage adopted in this work. The best solution is accepted if its difference with the optimal solution is less than or equal one percent.

$$|Bestcost - optimalcost| \leq 0.01 \tag{11}$$

Due to the randomness of the initial values and the selection of conditions, single run for the program is not a wise solution to assess the performance of each algorithm. For this reason, each algorithm is executed thirty times in this work. The success rate of each algorithm is given in (12), where the success rate (SR) is the percentage of the number of successful runs to the total number of runs.

$$SR = \frac{number\ of\ successful\ run}{number\ of\ run} \times 100\% \tag{12}$$

In Tables (II, III, IV, and V), the comparisons between the algorithms that have already been mentioned above are represented for dimensions (5, 10, 15, and 20) respectively. Through these tables, one can notice that the comparison was made on the basis of the statistical results represented by mean, standard deviation, best cost median, and success rate. These statistical calculations were done through the program (IBM SPSS Statistics 26) program version 20. The tests were carried out by using a laptop with Core i7, 2.4 GHz processor, Windows 10 Pro, a 512 Gb SSD and 16 Gb RAM.

After analyzing the statistical results, it is noted that the proposed CDOA provides an excellent performance compared to the other algorithms for the unimodal functions in all dimensions. For the multimodal function, the results were also acceptable and can be relied on for the dimensions less than or equal to 20. For dimensions larger than 20, the success rate is low in some functions. This is a real interpretation of the No Free Lunch theorem (NFL) [26] which stipulates that there is no algorithm that can solve all equations and engineering applications. Moreover, the algorithm can find the solution for dimensions larger than 20 to a certain type of the equation and fail in other types. It is worth mentioning here that the proposed algorithm does not match the dimensions of more than twenty variables of the multimodal functions, otherwise CDOA is very efficient and reliable. Among the properties of the algorithms that should be mentioned are the speed of response, speed of implementation, and speed of convergence. For convergence assessment purposes, Fig. 3. demonstrates the convergence of the proposed algorithm toward the optimum solution for some of the benchmark functions given previously. A few iterations are required for CDOA to converge to the optimal solution, and this is considered as one of the main advantages of this algorithm. V lists the time required to execute each algorithm. It is clear that CDOA speed of execution is comparable with the fast modified CA algorithm. Another important feature for the proposed algorithm can be noticed in this work. When looking at CDOA and comparing it with its counterparts, it can be seen that there is no parameter in this algorithm that need to be set to increase its convergence speed. In contrast, PSO requires four setting parameters namely individual learning factor, maximum velocity, inertia weight, and social learning factor. CSA requires setting adjustable parameters called flight, length, and awareness probability. Modified CA parameters that need to be set are the camel endurance and the camel visibility.

V. ENGINEERING APPLICATIONS

A. PID Controller

The term PID is an abbreviation of the words proportional, integral, and derivative. Consequently, the controller has three coefficients to be optimized one for the proportional gain, the other for the integration gain, and the third is for the derivative gain. Each coefficient has a certain effect on the system; for example, it is possible to improve the steady state error by controlling the proportional coefficient, or to eliminate the steady state error through the integration coefficient. In addition, the overshoot of the system is controlled through the derivative coefficient [40]. The structure of PID controller system is shown in, Fig. 4. The three coefficients of the PID controller ($K_P, K_i, and K_d$) given in (13) can be optimized to improve the performance of the entire system.

$$control\ signal = K_P e(t) + K_i \int_0^t e(t)dt + K_d \frac{d}{dt} e(t) \tag{13}$$

where $e(t)$ is the error signal that is corresponding to the difference between the input $r(t)$ and the output of the system $y(t)$. The main function of PID controller is to keep the value of the error signal as low as possible. As mentioned earlier, the error signal can be reduce by optimizing the controller coefficients $K_P, K_i, and K_d$. The transfer function $L(s)$ of the controller is given by [41]:

$$L(s) = \frac{K_d S^2 + K_p S + K_i}{S} \tag{14}$$

The open loop gain of the uncontrolled system is symbolized as $G(s)$, so the transfer function of closed loop system with PID controller is as given below:

$$G_{closeloop}(S) = \frac{L(S)G(S)}{1+L(S)G(S)} \tag{15}$$

In fact, the main goal of the PID controller is to obtain high accuracy of data in addition to the level of quality in the transient process [42]. Thus, the three main parameters that are used to describe the quality of the transient response any system are listed below [43]:

• Rising time t_r: The time required for the response change to reach 90% of its steady state value.

• Settling time t_s: the time required for the response to be within |2%| of the steady state value.

• Overshoot M_r (%) : the maximum value that the response reaches above the steady state value.

In this work, the CDOA is used to optimize the PID coefficients $K_P, K_i, andK_d$ to get the best transient response which has as minimum values of $t_r, t_s, andM_r$. The fitness function that is used to optimize the aforesaid three transient response parameters is as follows[41]:

$$F = min\left[(1-e^{-\alpha})(M_P - E_{SS}) + e^{-\alpha}(t_r + t_s)\right] \tag{16}$$

where α is the weight factor, and E_{SS} is the steady state error which denotes Laplace transform of the error signal $e(t)$ at the steady state ($t \to \infty$) as given in (17)[41].

$$E_{ss} = \lim_{S \mapsto 0} S \frac{R(s)}{1+G_{closeloop}} \tag{17}$$

where $R(s)$ is the desired set of input in S-domain. If the input is a unit step, its Laplace transform is equal to $\frac{1}{s}$. As a result, the steady state error becomes as in (18):

$$E_{ss} = \frac{1}{1+G_{closeloop}(0)} \tag{18}$$

The CDOA is applied to different standard systems, and the time response is demonstrated before and after the optimization process. The standard systems that were selected in this work are as follows [41]:

A. Second order system: $G(S) = \frac{20}{S^2+0.5S+10}$

B. Forth order system:

$G(S) = \frac{25.2S^2+21.2S+3}{S^4+16.4825S^3+23.8021S^2+14.8566S+10.2497}$

C.Fifth order system

$G(S) = \frac{25.2S^2+21.2S+3}{S^5+16.58S^4+25.41S^3+17.18S^2+11.7S+1}$

D. Forth order system with time delay

$G(S) = \frac{10}{S^4+10S^3+35S^2+50S+24} e^{-3S}$

E. Simple mass – damper system:

$G(S) = \frac{1}{S^2+10S+20}$

Based on the fitness function in (16) and by selecting α to be equal to 0.5 for each system, the unit step response after optimization results in the coefficient values given in Table VI: To know the performance of each system before and After applying the PID controller, Table VII gives a comparison of the time response before and after inserting the optimized controller, where the optimized system results in negligible values of rising time, settling time, and overshoot for all systems. Fig. 5. illustrates the unit step time response for each system before and after the optimization process.

B. Cantilever Beam

In mechanical engineering systems, there is an important element called the cantilever beam shown in. Fig. 6. From Table IX, it is noticed that the resulted best value of the fitness function in the proposed algorithm is less than that of other algorithms. This indicates the success of the algorithm in this application. The results can be seen in Table IX row No. 7, which lists the best values of the fitness functions for this application.

The main objective in this optimization process is to reduce the weight of the side beam. As shown in the figure, the cantilever beam contains five hallows in the form of square boxes. The lengths of the five boxes in this process are variable. The fitness function and its constraints for this problem are as in (19) and (20), respectively [44].

$$f(x) = 0.6224(x_1 + x_2 + x_3 + x_4 + x_5) \tag{19}$$

$$g(x) = \frac{61}{x_1^3} + \frac{27}{x_2^3} + \frac{19}{x_3^3} + \frac{7}{x_4^3} + \frac{1}{x_5^3} - 1 \leq 0 \tag{20}$$

where $x_1, x_2, x_3 x_4, andx_5$ represent the dimensions of the cantilever beam. The rang of these variables is $0.01 \leq x_i \leq 100$. Table IX gives the results of the optimization for this problem after applying the proposed CDOA algorithm, and then comparing it with particle swarm optimization (PSO), genetic algorithm (GA), multi-verse optimizer (MVO), water wave optimization (WWO), sine cosine algorithm (SCA), and whole optimization algorithm (WOA) whose results are listed in [45].

VI. CONCLUSION

A new algorithm based on the diffusion of the pitting corrosion on the metal surface has successfully been presented in this work. The oxidation-reduction chemical reactions and the Gibbs free energy equation have been utilized to describe the corrosion diffusion that emulates the searching mechanism

TABLE II.
PERFORMANCE COMPARISON FOR THE BEST COST OF THE SIX ALGORITHMS WITH 1000 ITERATIONS, 30 RUNS AND VARIABLE DIMENSION = 5

FUN#	Method	Best	Mean	Std.	Median	SR%
FUN1	CDOA.	8.91E-210	1.7740581E-56	4.0924500E-57	9.98E-97	100
	Modified CA	1.18E-29	8.06E-25	2.27E-24	9.04E-27	100
	FKE	3.456E-50	6.7632E-30	1.11E-28	7.9845E-29	100
	CSA	1.87E-20	2.28E-18	4.49E-18	7.21E-19	100
	PSO	0	0	0	0	100
	SCA	5.17E-08	0.00004	5.61E-05	6.13E-07	100
FUN2	CDOA.	1.75E-236	2.3733333E-59	1.29992820E-58	1.67E-111	100
	Modified CA	5.24E-27	1.12E-24	2.24E-24	1.24E-25	100
	FKE	7.5865E-29	3.78627E-27	3.332E-26	2.67E-26	100
	CSA	1.79E-20	2.08E-18	4.28E-18	9.18E-19	100
	PSO	0	0	0	0	100
	SCA	0.080856	0.080972	0.00036	0.091922	100
FUN3	CDOA.	6.37E-204	6.7520666E-57	3.68765899E-56	3.82E-99	100
	Modified CA	2.49E-23	7.46E-18	1.75E-17	6.45E-19	100
	FKE	1.389E-27	8.429E-20	8.765E-19	5.412E-22	100
	CSA	1.17E-21	1.28E-15	4.38E-15	5.47E-17	100
	PSO	0	0	0	0	100
	SCA	0.009435	0.0089954	0.00032	0.08864	85
FUN4	CDOA.	4.44E-15	9.7032666E-14	2.6550080E-13	7.99E-15	100
	Modified CA	3.29E-14	1.14E-11	2.26E-11	3.93E-12	100
	FKE	2.182E-15	2.751E-10	1.7193E-9	1.729E-9	100
	CSA	2.19E-09	9.39E-09	4.32E-09	8.75E-09	100
	PSO	4.35E-14	0.1142	0.3527	1.13E-12	93
	SCA	0.008321	0.00066785	4.02E-08	0.009953	70
FUN5	CDOA.	1.21E-189	7.1333333E-60	3.90708757E-59	1.12E-93	100
	Modified CA	0	0	0.0073	0.0099	60
	FKE	2.447E-121	5.193E-33	2.221E-28	3.629E-51	100
	CSA	0	0.0131	0.0107	0.0123	50
	PSO	0	0.0171	0.0171	0.016	50
	SCA	0.0071067	0.0071037	4.02E-06	0.062754	45
FUN6	CDOA.	5.56E-163	9.233333E-61	5.057304E-60	2.82E-89	100
	Modified CA	0	4.77E-09	2.61E-08	0	100
	FKE	0	1.188E-15	3.3664E-14	1.519E-14	100
	CSA	0	1.3598	0.9944	0.995	20
	PSO	0	0.4643	0.8152	0	67
	SCA	0.00005643	6.87695E-07	0.0000439	0.0006675	95

TABLE III.
PERFORMANCE COMPARISON FOR THE BEST COST OF THE SIX ALGORITHMS WITH 1000 ITERATIONS, 30 RUNS AND VARIABLE DIMENSION = 10

FUN#	Method	Best	Mean	Std.	Median	SR%
FUN1	CDOA.	8.34E-23	7.8456444E-07	4.0852034E-06	1.35E-17	100
	Modified CA	5.07E-20	1.60E-17	5.61E-17	2.07E-18	100
	FKE	4.822E-19	3.775E-18	4.662E-16	3.101E-17	100
	CSA	1.14E-19	1.70E-18	2.44E-18	8.03E-19	100
	PSO	0	0	0	0	100
	SCA	1.46E-07	2.70E-05	6.02E-05	2.57E-06	100
	SCA	9.15E-24	5.71E-21	9.86E-21	6.66E-23	100
FUN2	CDOA.	1.55E-23	8.7531616E-07	4.7830732E-06	2.47E-16	100
	Modified CA	3.27E-20	1.68E-17	2.61E-17	2.96E-18	100
	FKE	1.001E-19	2.311E-16	3.85E-16	4.1073E-17	100
	CSA	1.25E-19	2.73E-17	3.35E-18	7.12E-18	100
	PSO	0	0	0	0	100
	SCA	6.59325E-07	2.59838E-05	8.9E-05	8.69E-06	100
FUN3	CDOA.	2.23E-24	2.9200013E-06	0.00001	6.81E-19	100
	Modified CA	2.40E-16	4.34E-14	7.06E-14	1.43E-14	100
	FKE	7.926E-15	9.251E-13	1.395E-13	2.592E-13	100
	CSA	8.03E-12	1.24E-09	2.05E-09	3.26E-10	100
	PSO	0	0	0	0	100
	SCA	1.9732E-06	0.0000024	6.5E-07	2.976E-06	100
FUN4	CDOA.	1.07E-24	0.000013	0.00005	7.38E-15	100
	Modified CA	3.26E-10	2.72E-08	5.81E-08	1.36E-08	100
	FKE	1.096E-9	3.966E-07	4.27E-07	7.553E-07	100
	CSA	3.14E-06	8.25E-01	9.58E-01	5.73E-04	57
	PSO	4.44E-15	5.27E-15	1.53E-15	4.44E-15	100
	SCA	5.98E-05	2.46E-04	3.57E-04	6.87E-03	90
FUN5	CDOA.	7.58E-18	4.9596798E-06	0.00003	1.18E-12	100
	Modified CA	0	0.0583	0.0282	0.0253	50
	FKE	2.443E-12	2.467E-04	0.000024	0.00415	56.667
	CSA	1.24E-10	0.0416	0.0306	0.032	37
	PSO	0	0.0641	0.0374	0.0363	30
	SCA	-0.0071067	0.0071011	7.18E-06	-0.0099654	70
FUN6	CDOA.	2.73E-25	1.32000004E-08	4.3326666E-08	1.77E-18	100
	Modified CA	0	9.47E-09	5.19E-15	0	100
	FKE	4.592E-21	1.629E-06	8.264E-06	2.639E-14	100
	CSA	0	3.6482	1.7786	3.9798	10
	PSO	0	0.796	0.661	0.995	43
	SCA	0.00004598	0.005697	2.46E-03	0.0005699	65

TABLE IV.
PERFORMANCE COMPARISON FOR THE BEST COST OF THE SIX ALGORITHMS WITH 1000 ITERATIONS, 30 RUN AND VARIABLE DIMENSION = 15

FUN#	Method	Best	Mean	Std.	Median	SR%
FUN1	CDOA.	1.79E-18	1.7658937E-15	1.347340E-10	9.62E-14	100
	Modified CA	8.93E-17	7.0285433E-15	1.178094E-14	3.19E-15	100
	FKE	1.639E-17	8.3173492E-14	5.19412E-9	3.6142E-13	100
	CSA	5.80E-09	5.68E-08	3.68E-06	4.79E-08	100
	PSO	0	0	0	0	100
	SCA	0.00009671	0.0006719	0.00036	0.0001156	100
FUN2	CDOA.	1.46E-16	2.7790667E-07	6.150042E-07	3.73E-11	100
	Modified CA	9.52E-17	9.58720667E-15	1.583618E-14	3.25E-15	100
	FKE	4.1936E-14	7.206E-6	3.8412E-6	6.4196E-10	100
	CSA	2.57E-12	6.87E-11	6.55E-10	9.57E-11	100
	PSO	0	0	0	0	100
	SCA	0.00004872	0.0009475	0.001	0.0002257	100
FUN3	CDOA.	3.99E-21	5.6101150E-09	1.501606E-08	1.06E-11	100
	Modified CA	1.59E-17	4.4628E-16	1.101611E-15	1.14E-16	100
	FKE	2.8583E-19	2.4817E-08	5.1935E-07	2.741E-10	100
	CSA	9.67E-10	9.85E-09	8.71E-08	9.75E-09	100
	PSO	0	0	0	0	100
	SCA	0.0001157	0.0009846	0.00087	0.003741	100
FUN4	CDOA.	1.11E-21	3.537724082E-09	1.053506E-08	1.06E-12	100
	Modified CA	1.08E-15	0.00000000000026	2.813018E-13	1.63E-13	100
	FKE	2.0381E-20	5.281E-08	7.2481E-07	1.7429E-11	100
	CSA	6.15E-04	9.67E-01	9.75E-01	4.58E-03	46
	PSO	6.84E-13	7.35E-12	9.74E-12	5.48E-10	100
	SCA	2.25E-03	8.29E-03	9.17E-03	1.11E-02	83
FUN5	CDOA.	5.78E-16	0.0057	0.01319	3.50E-07	84
	Modified CA	3.66E-05	0.0754	0.04497	7.32E-02	43
	FKE	6.195E-08	0.0931	0.06342	9.420E-02	40
	CSA	3.46E-05	0.3357	0.5486	0.9745	30
	PSO	0	0.6485	0.7435	0.9974	27
	SCA	-0.06843	0.036419	2.21E-02	-0.77642	66
FUN6	CDOA.	1.60E-20	4.22325632E-08	1.077977E-07	5.59E-11	100
	Modified CA	1.24E-17	1.37837133E-14	2.434454E-14	2.57E-15	100
	FKE	2.052E-19	5.91302481E-08	4.719304E-06	7.019283E-10	100
	CSA	0	4.6547	2.0021	4.1297	6
	PSO	0	1.3479	1.27941	1.5697	33
	SCA	0.0044587	0.068721	6.57E-02	0.2347	56

TABLE V.
PERFORMANCE COMPARISON FOR THE BEST COST OF THE SIX ALGORITHMS WITH 1000 ITERATIONS, 30 RUNS AND VARIABLE DIMENSION = 20

FUN#	Method	Best	Mean	Std.	Median	SR%
FUN1	CDOA.	8.61E-13	4.343590E-06	0.00001	3.34E-06	100
	Modified CA	4.28E-13	0.000000000001	2.497574E-12	2.73E-13	100
	FKE	1.8274E-12	2.001E-10	8.162E-10	3.47E-12	100
	CSA	2.25E-08	5.48E-07	3.36E-05	6.79E-06	100
	PSO	2.536E-11	1.478E-09	1.5E-09	4.578E-10	100
	SCA	0.0005476	0.006578	0.00421	0.006715	100
FUN2	CDOA.	1.02E-18	3.1268824E-06	3.996994E-06	1.65E-06	100
	Modified CA	6.24E-14	0.0000000000015	1.803449E-12	8.40E-13	100
	FKE	9.4182E-12	1.390E-9	4.7726E-10	8.40E-11	100
	CSA	6.94E-11	4.72E-10	7.76E-09	6.48E-10	100
	PSO	6.55E-30	2.60E-28	1.64E-26	1.36E-26	100
	SCA	0.00099173	0.002846	0.00337	0.0067941	100
FUN3	CDOA.	1.89E-12	3.1365680E-08	4.576027E-08	1.82E-08	100
	Modified CA	2.14E-15	5.5209666E-14	9.923128E-14	3.65E-14	100
	FKE	1.0849E-13	7.5104E-12	8.6103E-12	8.889E-12	100
	CSA	6.49E-09	3.42E-08	3.80E-07	2.82E-08	100
	PSO	3.65E-21	3.50E-18	6.48E-17	6.48E-20	100
	SCA	0.0036749	0.0045679	0.06479	0.022457	93.33
FUN4	CDOA.	1.15E-11	0.0000033	4.576027E-08	2.78E-06	90
	Modified CA	4.45E-15	4.63056667E-14	3.445415E-14	4.87E-14	100
	FKE	3.591E-9	2.956E-4	8.4139E-06	5.827E-04	86.66
	CSA	2.59E-03	9.90E-01	1.00E+00	3.15E-02	40
	PSO	8.21E-13	9.88E-12	1.56E-11	6.21E-10	100
	SCA	6.48E-03	9.97E-03	1.21E-02	6.48E-02	70
FUN5	CDOA.	1.76E-11	3.0391017E-06	0.00000442	7.31E-07	100
	Modified CA	6.52E-05	0.0171	0.02164	1.07E-03	50
	FKE	1.184E-04	0.00247	0.003357	8.426E-03	56.66
	CSA	8.67E-04	0.57894	0.87614	1.00324	26
	PSO	0	0.8736	0.83201	1.0457	50
	SCA	-0.0088745	0.065134	7.30E-02	-0.0090247	53
FUN6	CDOA.	2.87E-08	0.0002	0.00042	1.00E-04	100
	Modified CA	1.21E-09	1.9891667E-08	2.401251E-08	1.38E-08	100
	FKE	9.0648E-08	4.7263E-07	1.9472E-07	7.5936E-06	100
	CSA	0.024567	5.0147	3.657	5.98461	0
	PSO	0	1.5367	1.4249	1.6794	30
	SCA	0.008674	0.07894	8.86E-02	0.30497	43

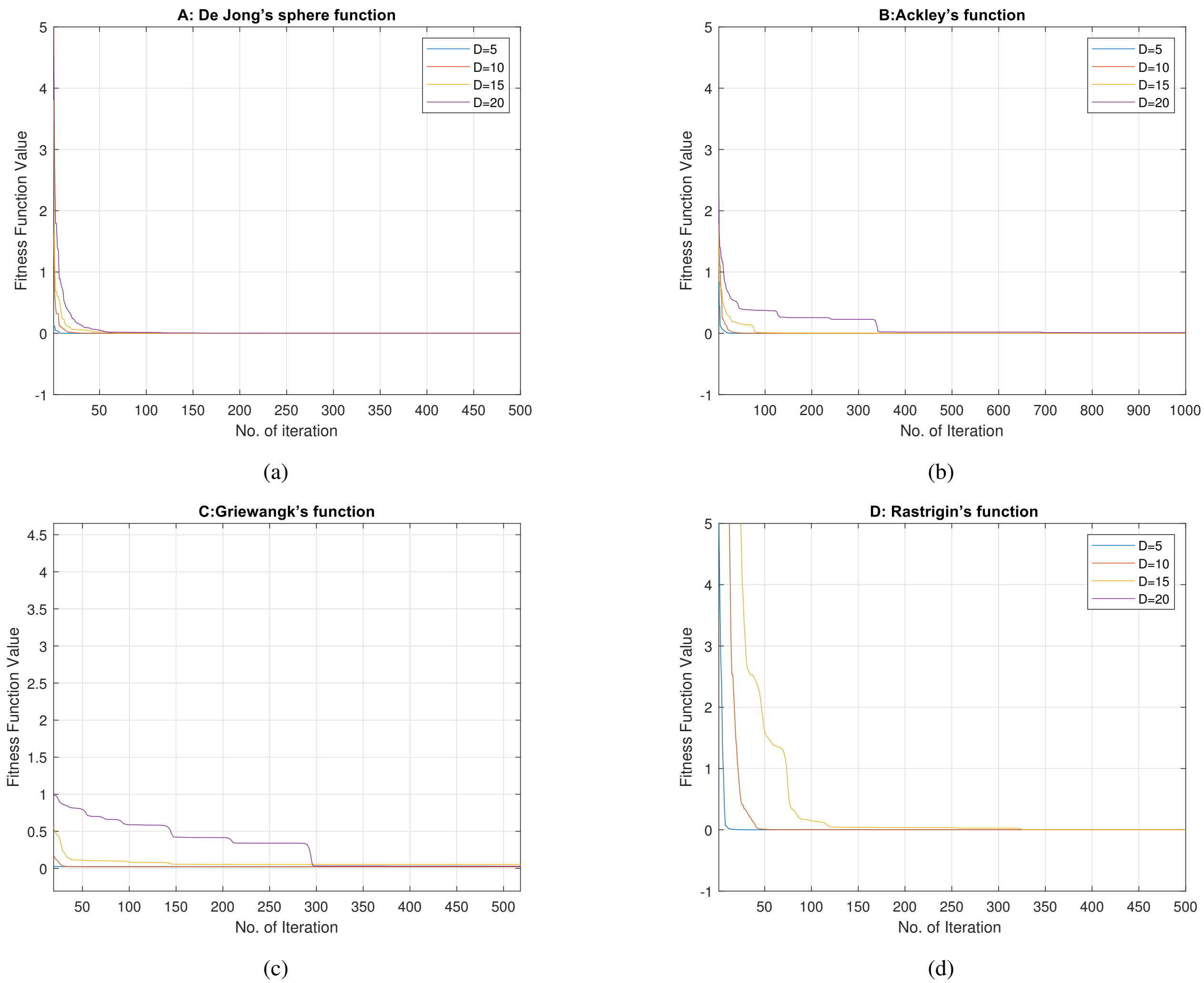

Fig. 3. curve of convergence (cost value with iteration) for different variable dimensions and different fitness function a: De Jong's sphere function, b: Ackley's function, c: Griewangk's function, and d: Rastrigin's function

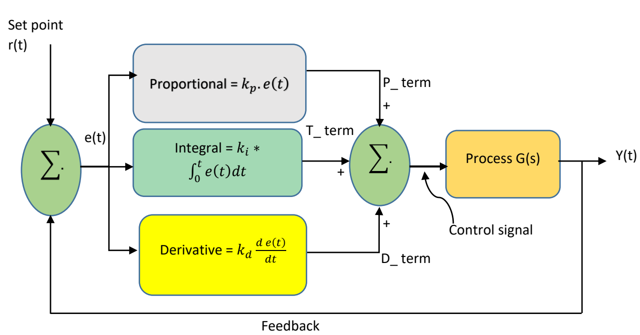

Fig. 4. PID controller of a closed loop system

of the proposed algorithm. With the aid of some unimodal and multimodal benchmark functions, it is found that the algorithm gives the best solution compared with PSO, CA, CSA, and SCA for the unimodal problems at any dimension for the optimization variable. However, the optimum solution of the proposed algorithm is the best compared to other solutions only for dimension of variable less than or equal to 20. For the PID kind of close loop control systems, the proposed algorithm provides a negligible amount of rise time, settling time, and overshoot. Finally, CDOA results in a minimum fitness function for the cantilever beam compared to PSO, GA, MVO, WWO, SCA, and WOA algorithms.

– **Future Works** As a future work, the chaos based initialization and pitting spreading will be used instead of the random

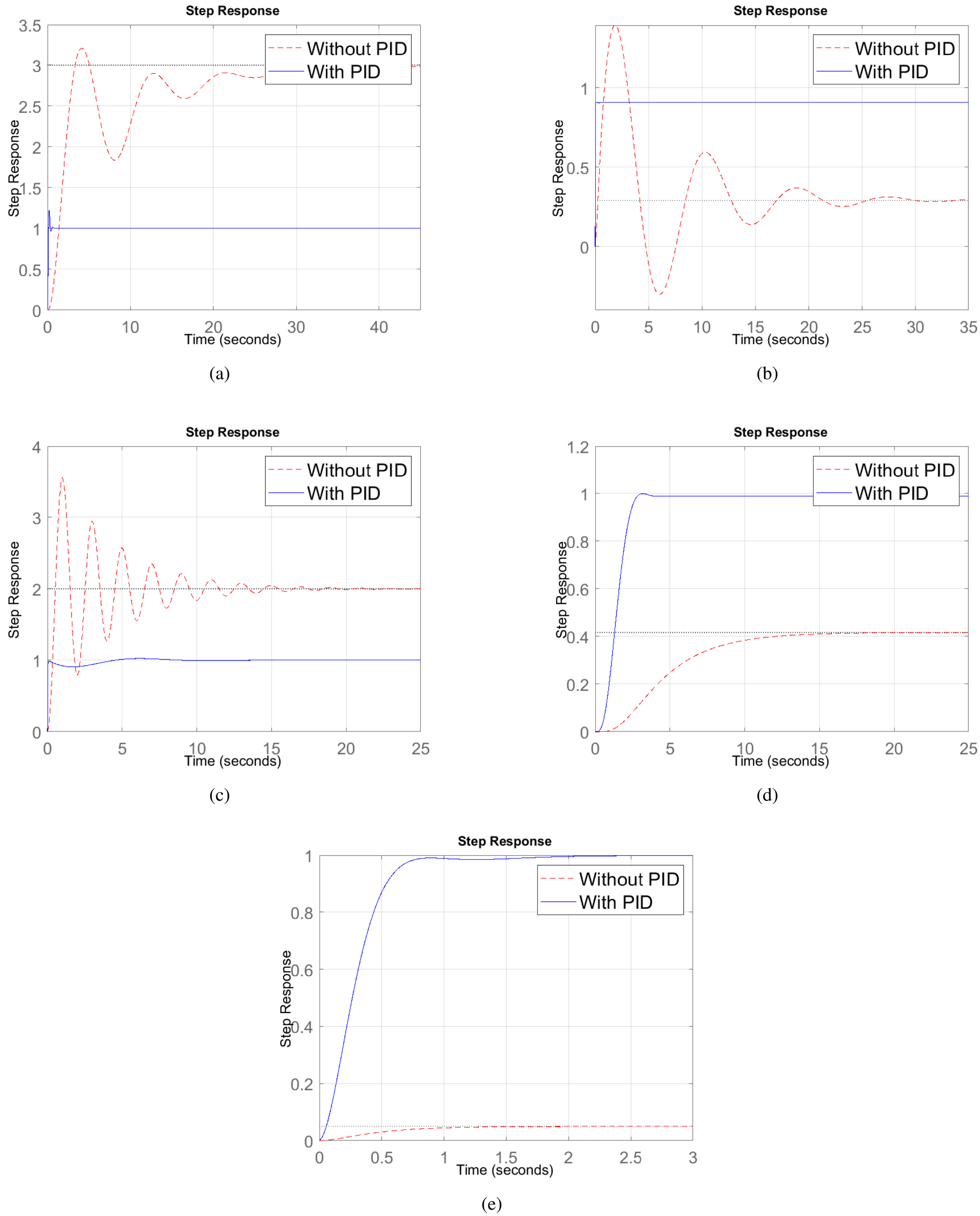

Fig. 5. The step response of different control system where are (a): system – A, (b): system – B, (c): system – C, (d): system – D, and (e): system – E.

TABLE VI.
EXECUTION TIME FOR EACH ALGORITHM WITH VARIABLE DIMENSION = 10 AND 1000 ITERATIONS

FUN#	Method	Time of 30 runs (s)	Av. Time of single run (s)
FUN1	CDOA.	8.182886	0.2827628
	Modified CA	5.026145	0.183246
	PSO	96.064317	3.281444
	SCA	30.9517536	1.398778
	FKE	120.453223	4.03333334
FUN2	CDOA.	8.237817	0.2755939
	Modified CA	5.51047	0.198275
	PSO	97.913156	3.352465
	SCA	42.2693127	1.422136
	FKE	130.6689	4.366667
FUN3	CDOA.	8.030139	0.2787653
	Modified CA	12.43114	0.43207
	PSO	100.251417	3.363865
	SCA	68.963951	3.336548
	FKE	111.99865	3.73333334
FUN4	CDOA.	8.146566	0.272987895
	Modified CA	7.569398	0.271435
	PSO	99.617142	3.436092
	SCA	73.741268	2.458223
	FKE	114.698453	3.833334
FUN5	CDOA.	8.158127	0.2749393
	Modified CA	8.344597	0.294662
	PSO	94.737667	3.20187
	SCA	69.785214	2.335697
	FKE	100.5437	3.366667
FUN6	CDOA.	8.172816	0.272876
	Modified CA	6.912578	0.247527
	PSO	89.546399	3.015565
	SCA	78.379128	2.6987426
	FKE	95.8775	3.2

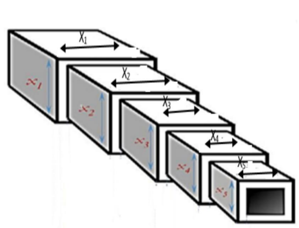

Fig. 6. Cantilever beam design problem

TABLE VII.
PID COEFFICIENTS FOR SYSTEM-A, SYSTEM-B, SYSTEM-C, SYSTEM-D, AND SYSTEM-E

System	Kp proportional coefficient	Ki integral coefficient	Kd derivative coefficient
System -A	10.54	11	6.223
System -B	2.2465	6.1281	6.5486
System -C	1.9558	4.1135	2.3717
System -D	4.8670	10.47404	2.6363
System -E	32.0583	0.0314	**68.8086**

dispersion. This may complicate the algorithm but may results in faster convergence to the optimal solution.

CONFLICT OF INTEREST

The authors have no conflict of relevant interest to this article.

TABLE VIII.
THE TRANSIENT PARAMETER WHEN WITHOUT PID CONTROLLER AND WITH PID CONTROLLER AFTER OPTIMIZED THE PROPORTIONAL COEFFICIENT, INTEGRAL COEFFICIENT, AND DERIVATIVE COEFFICIENT

#System	System-A		System-B		System-C		System-D		System-E	
	without PID	with PID	without PID	with PID	without PID	with PID	without PID	with PID	without PID	with PID
tr	0.3505	0.0099	0.1877	0.0129	2.1694	0.2339	7.3892	1.5683	0.8843	0.403
ts	15.1287	0.0175	24.5412	0.0229	33.5405	0.7756	14.1136	2.7102	1.5894	0.628
MP	77.9429	0.005	377.305	0	7.1052	0	0	1.80E-04	0	0

TABLE IX.
Compassion between the result of different algorithms for cantilever beam problem

Variables	PSO	GA	WWO	MVO	SCA	WOA	CDOA
X1	6.05099	6.03277	5.99823	6.05099	5.67787	5.78636	**6.0265**
X2	4.93196	5.31962	5.24612	4.93196	5.33850	5.57995	**4.8966**
X3	5.2118	4.48431	4.49356	5.21118	4.86170	4.28758	**4.5686**
X4	3.94183	3.48147	3.60865	3.94183	3.45494	3.74891	**3.5013**
X5	1.88577	2.15591	2.13733	1.88577	2.30102	2.16705	**2.0950**
F(x)	1.37063	1.33655	1.33702	1.37063	1.34650	1.34251	**1.31251**

REFERENCES

[1] I. Fister Jr, X.-S. Yang, I. Fister, J. Brest, and D. Fister, “A brief review of nature-inspired algorithms for optimization,” *arXiv preprint arXiv:1307.4186*, 2013.

[2] C. Blum, “Ant colony optimization: Introduction and recent trends,” *Physics of Life reviews*, vol. 2, no. 4, pp. 353–373, 2005.

[3] X.-S. Yang, “Firefly algorithm, stochastic test functions and design optimisation,” *International journal of bio-inspired computation*, vol. 2, no. 2, pp. 78–84, 2010.

[4] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in *Proceedings of ICNN’95-international conference on neural networks*, vol. 4, pp. 1942–1948, ieee, 1995.

[5] X.-S. Yang and S. Deb, “Engineering optimisation by cuckoo search,” *International Journal of Mathematical Modelling and Numerical Optimisation*, vol. 1, no. 4, pp. 330–343, 2010.

[6] C. Zhao and Y. Zhou, “A complex encoding flower pollination algorithm for global numerical optimization,” in *Intelligent Computing Theories and Application: 12th International Conference, ICIC 2016, Lanzhou, China, August 2-5, 2016, Proceedings, Part I 12*, pp. 667–678, Springer, 2016.

[7] M. Azizi, S. Talatahari, N. Khodadadi, and P. Sareh, “Multiobjective atomic orbital search (moaos) for global and engineering design optimization,” *IEEE Access*, vol. 10, pp. 67727–67746, 2022.

[8] Z. Zandi, E. Afjei, and M. Sedighizadeh, “Reactive power dispatch using big bang-big crunch optimization algorithm for voltage stability enhancement,” in *2012 IEEE International Conference on Power and Energy (PECon)*, pp. 239–244, IEEE, 2012.

[9] R. Formato, “Central force optimization: a new metaheuristic with applications in applied electromagnetics,” *Progress in electromagnetics research*, vol. 77, pp. 425–491, 2007.

[10] G. Zhao, X. Wang, H. Zhao, and Z. Jiang, “An improved pedestrian dead reckoning algorithm based on smartphone built-in mems sensors,” *AEU-International Journal of Electronics and Communications*, vol. 168, p. 154674, 2023.

[11] R. S. Ali, F. M. Alnahwi, and A. S. Abdullah, “A modified camel travelling behaviour algorithm for engineering applications,” *Australian Journal of Electrical and Electronics Engineering*, vol. 16, no. 3, pp. 176–186, 2019.

[12] T.-C. Chen, P.-W. Tsai, S.-C. Chu, and J.-S. Pan, “A novel optimization approach: bacterial-ga foraging,” in

Second international conference on innovative computing, Informatio and Control (ICICIC 2007), pp. 391–391, IEEE, 2007.

[13] X.-S. Yang, "A new metaheuristic bat-inspired algorithm," in *Nature inspired cooperative strategies for optimization (NICSO 2010)*, pp. 65–74, Springer, 2010.

[14] D. Teodorovic and M. Dell'Orco, "Bee colony optimization–a cooperative learning approach to complex transportation problems," *Advanced OR and AI methods in transportation*, vol. 51, p. 60, 2005.

[15] X.-S. Yang, J. M. Lees, and C. T. Morley, "Application of virtual ant algorithms in the optimization of cfrp shear strengthened precracked structures," in *International Conference on Computational Science*, pp. 834–837, Springer, 2006.

[16] F. M. Alnahwi, Y. I. Al-Yasir, D. Sattar, R. S. Ali, C. H. See, and R. A. Abd-Alhameed, "A new optimization algorithm based on the fungi kingdom expansion behavior for antenna applications," *Electronics*, vol. 10, no. 17, p. 2057, 2021.

[17] D. Simon, "Biogeography-based optimization," *IEEE transactions on evolutionary computation*, vol. 12, no. 6, pp. 702–713, 2008.

[18] Y. Shi, "An optimization algorithm based on brainstorming process," in *Emerging Research on Swarm Intelligence and Algorithm Optimization*, pp. 1–35, IGI Global, 2015.

[19] C. J. Bastos Filho, F. B. de Lima Neto, A. J. Lins, A. I. Nascimento, and M. P. Lima, "A novel search algorithm based on fish school behavior," in *2008 IEEE international conference on systems, man and cybernetics*, pp. 2646–2651, IEEE, 2008.

[20] M. M. Eusuff and K. E. Lansey, "Optimization of water distribution network design using the shuffled frog leaping algorithm," *Journal of Water Resources planning and management*, vol. 129, no. 3, pp. 210–225, 2003.

[21] A. Hatamlou, "Black hole: A new heuristic optimization approach for data clustering," *Information sciences*, vol. 222, pp. 175–184, 2013.

[22] A. Kaveh and S. Talatahari, "A novel heuristic optimization method: charged system search," *Acta mechanica*, vol. 213, no. 3, pp. 267–289, 2010.

[23] H. Eskandar, A. Sadollah, A. Bahreininejad, and M. Hamdi, "Water cycle algorithm–a novel metaheuristic optimization method for solving constrained engineering optimization problems," *Computers & Structures*, vol. 110, pp. 151–166, 2012.

[24] H. Shayeghi and J. Dadashpour, "Anarchic society optimization based pid control of an automatic voltage regulator (avr) system," *Electrical and electronic engineering*, vol. 2, no. 4, pp. 199–207, 2012.

[25] P. Civicioglu, "Artificial cooperative search algorithm for numerical optimization problems," *Information Sciences*, vol. 229, pp. 58–76, 2013.

[26] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE transactions on evolutionary computation*, vol. 1, no. 1, pp. 67–82, 1997.

[27] C. M. Hansson, "The impact of corrosion on society," *Metallurgical and Materials Transactions A*, vol. 42, pp. 2952–2962, 2011.

[28] Z. Szklarska-Smialowska and ZS-Smialowska, *Pitting and crevice corrosion*, vol. 446. NACE international Houston, TX, 2005.

[29] C. Andrade and C. Alonso, "Corrosion rate monitoring in the laboratory and on-site," *Construction and building materials*, vol. 10, no. 5, pp. 315–328, 1996.

[30] M. I. I. Bin, "Computational model of pitting corrosion," 2013.

[31] Z. Ahmad, *Principles of corrosion engineering and corrosion control*. Elsevier, 2006.

[32] N. Perez, *Electrochemistry and corrosion science*. Springer, 2004.

[33] E. E. Stansbury and R. A. Buchanan, *Fundamentals of electrochemical corrosion*. ASM international, 2000.

[34] M. G. Fontana, N. D. Greene, *et al.*, *Corrosion engineering*. McGraw-hill, 2018.

[35] G. Frankel, "Pitting corrosion," 2003.

[36] H. Kaesche, *Corrosion of metals: physicochemical principles and current problems*. Springer Science & Business Media, 2012.

[37] M. Molga and C. Smutnicki, "Test functions for optimization needs," *Test functions for optimization needs*, vol. 101, p. 48, 2005.

[38] A. G. H. et al, "Tcrow search algorithm: theory, recent advances, and applications," *IEEE Access*, vol. 8, pp. 173548–173565, 2020.

[39] A. I. Hafez, H. M. Zawbaa, E. Emary, and A. E. Hassanien, "Sine cosine optimization algorithm for feature selection," in *2016 international symposium on innovations in intelligent systems and applications (INISTA)*, pp. 1–5, IEEE, 2016.

[40] K. H. Raut and S. Vaishnav, "Performance analysis of pid tuning techniques based on time response specification," *International Journal of Innovative Research in Electrical, Electronics, Instrumentation & Control Engineering*, vol. 2, no. 1, 2014.

[41] I. Juniku and P. Marango, "Pid design with bio-inspired intelligent algorithms for high order systems," *International Journal of Mathematics and Computers in Simulation*, vol. 9, pp. 44–52, 2015.

[42] L. Shirokov, P. Chelyshkov, and E. Romanenko, "Automated management of engineering infrastructure of pools of different function," in *MATEC Web of Conferences*, vol. 86, p. 04062, EDP Sciences, 2016.

[43] I. Juniku and P. Marango, "A comparison of pso and bfo applications for the pid controller synthesis in time-delay systems,"

[44] S. Mirjalili, S. M. Mirjalili, and A. Hatamlou, "Multi-verse optimizer: a nature-inspired algorithm for global optimization," *Neural Computing and Applications*, vol. 27, pp. 495–513, 2016.

[45] "https://www3.ntu.edu.sg/home/epnsugan/," 2015.