Engineering Science and Technology, an International Journal 56 (2024) 101765

Contents lists available at ScienceDirect

Engineering Science and Technology, an International Journal

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

Full Length Article

Hybrid coati–grey wolf optimization with application to tuning linear quadratic regulator controller of active suspension systems ☆

Hasan Başak

Artvin Coruh university, Faculty of Engineering, Electrical-Electronics Engineering, Seyitler Campus 08100, Artvin, Turkey

ARTICLE INFO

Keywords:
Active suspension control
Coati optimization algorithm (COA)
Grey wolf optimizer (GWO)
Hybrid optimization
Swarm intelligence

ABSTRACT

Vehicle suspension systems have become increasingly crucial for both driving safety and comfort. Active suspension systems can dynamically adjust suspension characteristics in real-time by introducing force into the system. Designing a controller for the real-time adjustment of the control force in active suspension systems is essential to meet challenging control objectives, including body acceleration, suspension deflection, and tire deflection. This article proposes a hybrid optimization approach named Coati–Grey Wolf Optimization (COAGWO), which combines the strengths of the coati optimization algorithm and grey wolf optimization to tune the gains of linear quadratic control applied to vehicle suspension systems. The COAGWO algorithm incorporates a unique strategy inspired by the Coati Optimization Algorithm, allowing wolves to climb trees. This enhancement significantly improves the wolves' ability to explore the global search space and reduces the likelihood of being trapped in local optima. Initially, we conduct extensive experiments using a suite of challenging optimization problems from the CEC2019 benchmark to evaluate the effectiveness of the COAGWO algorithm. The effectiveness of COAGWO is compared against several state-of-the-art algorithms, including grey wolf, coati, aquila-grey wolf, whale, reptile search, tunicate swarm, and seagull optimization algorithms. The experimental results demonstrate that COAGWO consistently outperforms these algorithms in terms of solution quality and convergence speed. For the optimal weight selection problem of linear quadratic control applied to the control of vehicle suspension systems, the excellent performance of the proposed method is illustrated through comparative simulation studies under various road disturbance conditions. The results indicate that the COAGWO algorithm achieves a more efficient active suspension system compared to competitor algorithms by reducing the overall acceleration of the driver's body, thereby enhancing ride comfort.

1. Introduction

Active suspension systems in vehicles aim to ensure ride comfort, road holding, and passenger safety across varying road irregularities. These systems dynamically adjust the suspension settings in real-time to adapt to changing road conditions, thereby enhancing vehicle performance and passenger experience. To fully leverage the potential of active suspension systems, the control algorithms must effectively manage and respond to diverse and unpredictable road profiles. In the literature, various control methods have been designed for active suspension systems, such as sliding mode control [1], adaptive control [2], fuzzy H_∞ control [3], fuzzy [4], and PID with the genetic algorithm [5]. The Linear Quadratic Regulator (LQR) controller is designed to obtain optimal performance without deteriorating conflicting design requirements [6–8]. Guaranteed stability, robustness, and a systematic design approach are provided by the LQR controller for multiple-input multiple-output systems. To compute an optimal state feedback gain, a quadratic performance index, comprising weighted state and input variables, is minimized. In spite of the potential advantages, one of the difficulties in LQR design is the optimal selection of the weighting matrices, which does not have efficient procedure. Bryson's method [9] can be used to obtain the initial selection of weighting matrices. However, this method is a trial-and-error process that is time-consuming and tiring. Meta-heuristic algorithms have recently emerged as an alternative tuning mechanism to address the selection of the weighting matrices. These algorithms, inspired by natural phenomena and biological processes, offer a flexible and efficient approach to optimizing complex systems. By leveraging principles such as population-based search, evolutionary strategies, and swarm intelligence, meta-heuristic algorithms can effectively explore solution spaces and find near-optimal solutions even in highly nonlinear and dynamic environments. In the literature, several researchers have proposed swarm intelligence methods, as listed in Table 1, to determine optimal weighting matrices.

☆ Peer review under responsibility of Karabuk University.

E-mail address: hasanbasak@artvin.edu.tr.

https://doi.org/10.1016/j.jestch.2024.101765
Received 9 February 2024; Received in revised form 7 June 2024; Accepted 7 July 2024
Available online 23 July 2024

Table 1
Recent LQR tuning efforts.

Algorithm	Application	Reference
ACO	Active suspension system	[7]
AIWPSO	Active suspension system	[8]
PSO	DC motor system	[10]
QPSO	Aircraft landing flare system	[11]
PSO	Overhead crane system	[12]
BA	Active suspension system	[13]
GWO	Active suspension system	[13]
AFSA	Active suspension system	[14]
APPO	Active suspension system	[15]

The primary question in the study of meta-heuristic algorithms is whether, given the numerous optimization algorithms that have been developed to date, there is still a need to introduce new methods. In response to this question, the No Free Lunch (NFL) theorem [16] states that the efficacy of an algorithm in addressing a specific set of optimization problems does not guarantee the same performance in other optimization issues. The NFL theorem thus motivates authors to design newer algorithms to find better solutions to optimization problems. In the last two decades, many meta-heuristics have been introduced with various characteristics, such as the Particle Swarm Optimization (PSO) algorithm [17], Genetic Algorithm (GA) [18], Ant Colony Optimization (ACO) [19], and recently introduced algorithms such as Bacterial Foraging Optimization (BFO) algorithm [20], Firefly Algorithm (FA) [21], Gravitational Search Algorithm (GSA) [22], Slime Mould Algorithm (SMA) [23], and Harris Hawks Optimization (HHO) algorithm [24]. These algorithms successfully solve many complex problems, such as resource planning issues [25], engineering optimization [26], estimation of energy usage models [27], graphic processing [28], handwritten word recognition [29], health information categorization [30–32], economic forecasting [33,34], and issues in the education field [35]. However, these algorithms have complex structures due to having many adjustment parameters.

The Grey Wolf Optimizer (GWO) [36] is a metaheuristic optimization algorithm inspired by the hunting behavior of wolf packs. It has some advantages, such as having few parameters and implementation flexibility [37]. Therefore, scholars have paid attention to the GWO algorithm to apply it efficiently in different fields such as the reactive power dispatch problem [38], streamflow forecasting [39], image segmentation [40], Proportional Derivative Integral (PID) control of a brushless direct current motor [41], non-integer order PID scheme of a direct current motor [42], and so on. GWO demonstrates good performance compared with traditional methods, but in some cases, there are disadvantages such as slow convergence, inaccuracy, and failure to find the global optimum. To overcome these problems, different improvements have been proposed by the authors. For example, Dhargupta et al. [43] enhanced the effectiveness of the GWO algorithm using the selective opposition method. Purushothaman et al. [44] presented a fusion of GWO and Grasshopper Optimization in their hybrid method. Mohammed et al. [45] hybridized GWO with the whale optimization algorithm for global numerical optimization. GWO based on the Aquila Optimization algorithm was presented in [46].

Hybrid optimization algorithms successfully combine suitable optimization algorithms, exhibiting good performance without trapping in local optima and having a fast convergence rate. This paper proposes a hybrid optimization algorithm (COAGWO) constructed via the Grey Wolf Optimizer (GWO) and the Coati Optimization Algorithm (COA) [47]. The COAGWO algorithm simulates a hybrid strategy similar to that employed by coatis when attacking iguanas, as well as the hunting behavior of wolves. The proposed hybrid algorithm improves the balance between exploration and exploitation in attempting to find a global solution. It has been observed that the performance of COAGWO is better than that of both the GWO and COA algorithms. The key contributions of this study are summarized as follows:

- Due to the shortcomings of the original GWO algorithm, we give the wolves the ability to climb trees like coatis. By integrating the exploration features of the COA algorithm, we enhance its exploration ability, leading to improved convergence accuracy.
- The effectiveness of the proposed COAGWO algorithm is validated using the CEC2019 benchmark problems. Comparative performance evaluation of the proposed hybrid algorithm with seven other algorithms confirms that COAGWO has efficient convergence and exploration abilities.
- The Wilcoxon rank-sum test and the ANOVA test are used to assess the statistical significance of the performance achieved by the proposed COAGWO algorithm.
- The proposed algorithm addresses the weight selection problem inherent in the Linear Quadratic Regulator (LQR) controller for a quarter-car active suspension system. Simulation studies are conducted to demonstrate the effectiveness of this method.
- The COAGWO, GWO, and COA algorithms are employed to determine weighting matrices for LQR controllers, followed by a comparison of vehicle body acceleration responses. The COAGWO algorithm demonstrates enhanced convergence speed and accuracy compared to the GWO and COA algorithms in determining the weighting matrices for the LQR controller in the active suspension system.
- The proposed hybrid algorithm is also compared with published LQR controllers based on GWO, Bat Algorithm (BA), Ant Colony Optimization (ACO), Adaptive Inertial Weight Particle Swarm Optimization (AIWPSO), and Adaptive Prey-Predictor Optimization (APPO) algorithms. The comparison focuses on vehicle body acceleration, suspension deflection, and tire deflection.

The remainder of the paper is organized as follows: Section 2 provides a comprehensive explanation of our proposed approach, COAGWO. Experimental validation of the proposed COAGWO approach is performed utilizing the CEC2019 benchmark problems in Section 3. The proposed COAGWO algorithm is applied to tuning the LQR controller for a quarter-car active suspension system in Section 4. Simulation results are presented in Section 5. Section 6 discusses the validation techniques applied to ensure the reliability of the proposed method. Finally, Section 7 presents conclusions and provides some future directions.

2. Proposed hybrid COAGWO algorithm

In this section, we focus on developing a hybrid method by combining the features of the GWO and COA algorithms. We assume that grey wolves can identify their prey and attack them. To enhance their capabilities, we give grey wolves the ability to climb trees like coatis. Therefore, grey wolves are referred to as coati-wolves in the proposed algorithm. In this strategy, a group of coati-wolves climbs the tree to reach the prey and scare it, while another group waits under the tree until the prey falls to the ground. The coati-wolves then attack the fallen prey and hunt it. This approach enables the coati-wolves to move in different directions in the search space, enhancing the algorithm's exploration ability in problem-solving space. The pattern diagram of this strategy in the proposed algorithm is displayed in Fig. 1. To present the mathematical model of the proposed COAGWO, the population of the algorithm is initialized as follows:

$$\begin{bmatrix} S_1 \\ \vdots \\ S_i \\ \vdots \\ S_N \end{bmatrix}_{N\times 1} = \begin{bmatrix} s_{1,1} & \cdots & s_{1,j} & \cdots & s_{1,m} \\ \vdots & \ddots & \vdots & \iddots & \vdots \\ s_{i,1} & \cdots & s_{i,j} & \cdots & s_{i,m} \\ \vdots & \iddots & \vdots & \ddots & \vdots \\ s_{N,1} & \cdots & s_{N,j} & \cdots & s_{N,m} \end{bmatrix}_{N\times m} \tag{1}$$

with

$$s_{i,j} = LB_j + r(UB_j - LB_j), \quad i = 1, 2, \ldots, N, \quad j = 1, 2, \ldots, m \tag{2}$$

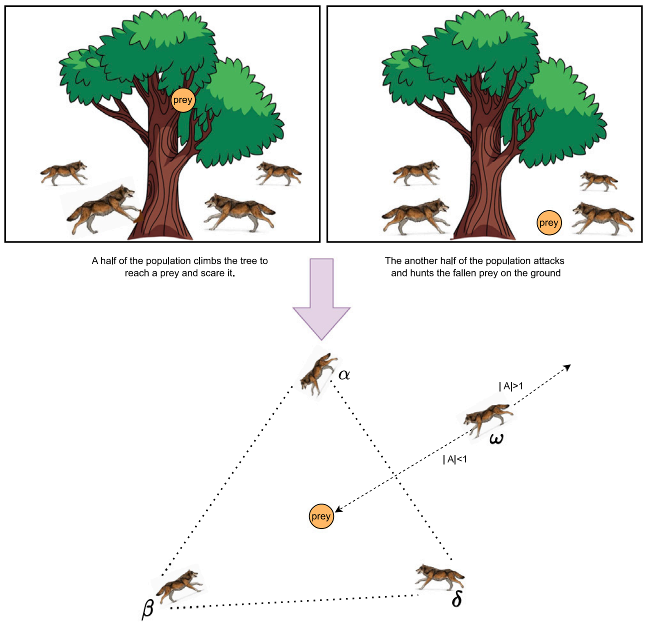


Fig. 1. Pattern diagram of the proposed COAGWO.

In which S_i represents the location of the ith coati-wolf, $s_{i,j}$ denotes the value for the jth decision variable proposed by the ith coati-wolf, N is the number of the population, m is the number of decision variables, $r \in [0,1]$ is a random real number, LB_j and UB_j are the lower and upper bounds of the jth decision variable, respectively. Now, each coati-wolf is considered as a potential solution, and the objective function is evaluated using their values for problem variables.

$$F = \begin{bmatrix} F_1 \\ \vdots \\ F_i \\ \vdots \\ F_N \end{bmatrix}_{N\times 1} = \begin{bmatrix} F(S_1) \\ \vdots \\ F(S_i) \\ \vdots \\ F(S_N) \end{bmatrix}_{N\times 1} \tag{3}$$

where F is the vector of obtained objective function values for all coati-wolves, and F_i represents the objective function value for the ith coati-wolf.

Exploration phase 1: Assuming that half of the coati-wolves climb the tree while the other half observes the prey until it drops to the ground, we provide a mathematical simulation of the coati-wolves' behavior as they emerge from the tree as follows:

$$S_i^{P1} : s_{i,j}^{P1} = s_{i,j} + r.(s_{p,j} - I.s_{i,j}), \quad \text{for } i = 1,2,\ldots,\frac{N}{2} \text{ and} j = 1,2,\ldots,m. \tag{4}$$

The prey is dropped to the ground and then positioned randomly in the search area. Coati-wolves explore the search space on the ground, which is simulated using Eqs. (5) and (6):

$$S_p^G : s_{p,j}^G = LB_j + r.(UB_j - LB_j), j = 1,2,\ldots,m, \tag{5}$$

$$S_i^{P1} : s_{i,j}^{P1} = \begin{cases} s_{i,j} + r.(s_{p,j}^G - I.s_{i,j}), F_{s_{p,j}^G} < F_i \\ s_{i,j} + r.(s_{i,j} - s_{p,j}^G), \quad else, \end{cases}$$

$$for\ i = \frac{N}{2}+1,\ \frac{N}{2}+2,\ldots,N \text{ and } j = 1,2,\ldots,m \tag{6}$$

During the update process, if the objective function shows improvement, each coati-wolf moves towards the new location; otherwise, the coati-wolf stays at the current position. This update condition is simulated for each coati-wolf $i = 1,2,\ldots,N$ by

$$S_i = \begin{cases} S_i^{P1}, F_i^{P1} < F_i \\ S_i,\ else. \end{cases} \tag{7}$$

where S_i^{P1} represents the new location calculated for the ith coati-wolf, $s_{i,j}^{P1}$ denotes its jth dimension, F_i^{P1} represents its objective function value. S_p^G denotes the prey's position in the search space, $s_{p,j}^G$ refers to its jth dimension, and $F_{s_{p,j}^G}$ is its value of the objective function. Additionally, I is randomly chosen from $\{1,2\}$.

Exploration phase 2:

$$U = |\bar{C}s_{p,j} - s_{i,j}| \tag{8}$$

$$s_{i,j} = s_{p,j} - \bar{A}U \tag{9}$$

where $s_{i,j}$ represents the location vector of the coati-wolf, and $S_p(t)$ is the vector representing the position of the prey,

$$S_p(t) = \begin{bmatrix} s_{p,1} & \ldots & s_{p,j} \end{bmatrix} \tag{10}$$

in which $s_{p,j}$ is its jth dimension where

$$\bar{A} = 2a \times r_1 - a \text{ and } \bar{C} = 2r_2 \tag{11}$$

where $r_1 \in [0,1]$ and $r_2 \in [0,1]$ are random numbers, a is determined as:

$$a = 2 - \frac{2t}{T} \tag{12}$$

where T is the upper limit of the iterations. The positions of members in the population are updated by α, β, and δ coati-wolves. The hunting behavior of the coati-wolves is mathematically simulated as follows:

$$s_{i,j}^{cw1} = s_j^{\alpha} - \bar{A}_1 \left|\bar{C}_1 s_j^{\alpha} - s_{i,j}\right| \tag{13}$$

$$s_{i,j}^{cw2} = s_j^{\beta} - \bar{A}_2 \left|\bar{C}_2 s_j^{\beta} - s_{i,j}\right| \tag{14}$$

$$s_{i,j}^{cw3} = s_j^{\delta} - \bar{A}_3 \left|\bar{C}_3 s_j^{\delta} - s_{i,j}\right| \tag{15}$$

where coefficients of $\bar{A}_i$ and $\bar{C}_i$, $i = 1,2,3$ are similar to $\bar{A}$ and $\bar{C}$, respectively.

Exploitation phase: The exploitation phase looks for the best solutions based on available knowledge. This includes concentrating on the best-known solution areas and making further improvements in these areas. In this way, the algorithm can make further improvements to the optimization problem. When the movement of prey stops, it is attacked by the coati-wolves. As can seen from (11), the value range of $\bar{A}_1, \bar{A}_2$ and $\bar{A}_3$ are $[-2a, 2a]$. When a becomes a small value, the variation of $\bar{A}$ decreases. when $|\bar{A}| < 1$, exploitation of the algorithm starts. Eq. (16) describes the exploitation phase of the optimization problem.

$$s_{i,j} = \frac{s_{i,j}^{cw1} + s_{i,j}^{cw2} + s_{i,j}^{cw3}}{3} \tag{16}$$

Pseudo-code and flowchart of our COAGWO algorithm are given in Algorithm 1 and Fig. 2 respectively.

2.1. Computational complexity

Computational complexity estimates how the number of basic operations performed by an algorithm increases with the number of items in the dataset. Yan et al. [48] confirmed that the computational complexity of GWO concludes as follows:

$$O(NTm) \tag{17}$$

where N is the number of wolves and m is the number of problem variables, T is the number of iterations of the algorithm. Also, Dehghani et al. [47] affirmed that the total computational complexity of COA is

$$O(Nm) \times (1 + \frac{5T}{2}) \tag{18}$$

where N is the number of coatis and m is present as the dimensions.

Table 2
Details of CEC2019 benchmark problems.

No	Function name	Dimension	Search range	F_{min}
F01	*Storn's Chebyshev Polynomial*	9	[−8192, 8192]	1
F02	*Inverse Hilbert Matrix*	16	[−16 384, 16 384]	1
F03	*Lennard-Jones Minimum Energy Cluster*	18	[−4, 4]	1
F04	*Rastrigin's*	10	[−100, 100]	1
F05	*Griewangk's*	10	[−100, 100]	1
F06	*Weierstrass*	10	[−100, 100]	1
F07	*Modified Scwefel's*	10	[−100, 100]	1
F08	*Expanded Schaffer's F6*	10	[−100, 100]	1
F09	*Happy Cat*	10	[−100, 100]	1
F10	*Ackley*	10	[−100, 100]	1

Computational complexity of the proposed COAGWO algorithm can be defined as (i) Population initialization is equal to $O(Nm)$ where N is the number of coati-wolves and m is the number of problem variables, (ii) The process of updating the COAGWO population in the first phase has a computational complexity equal to $O(NmT)$, (iii) The process of calculating the prey's random position on the ground and calculating its objective function has a complexity equal to $O(NmT/2)$. (iv) The second phase (main GWO) of the process of updating the position of the coati-wolves has a computational complexity equal to $O(NmT)$. Thus, the total computational complexity of the proposed COAGWO is equal to

$$O(Nm) \times (1 + \frac{5T}{2}) \tag{19}$$

It can be concluded that the computational complexity of COAGWO still shows the same order time complexity as those of COA and GWO.

Algorithm 1 Pseudo-code of the Proposed COAGWO

```
Determine the maximum iterations, T
Define the population size for Coati-wolves, N
procedure COAGWO
    Initialize position of coati-wolf using Eq. (2).
    Evaluate the corresponding objective function values of each search agent. S_α, S_β, and S_β are the best, second best, and third best search agents respectively.
loop:
    for For iter = 1 : T do
        for For i = 1 : N/2 do
            Phase 1: Attacking strategy on the prey (Exploration Phase 1)
            Update the current location of the ith coati-wolf of the algorithm using (4)
            Find a new location of ith coati-wolf using
        end for i = 1 : N/2
        for For i = 1 + N/2 : N do
            Compute random position for the prey using Eq. (5)
            Compute new location of the ith member of the algorithm using Eq. (6)
            Find the location of ith coati-wolf using
        end for For i = 1 + N/2 : N
        Phase 2: Attacking strategy on the prey (Exploration Phase 2)
        for For i = 1 : N do
            Update Ā, C̄ and a
            Find the location of coati-wolf α, β and δ using Eqs.(13-15)
            Calculate new location of the wolves using Eq. (16) (Exploitation Phase)
        end for For i = 1 : N
    end for iter = 1 : T
    Output of the best solution obtained by COAGWO for a given problem.
end procedure
```

3. Experimental results on CEC-C06 2019 benchmark problems

n this section, we adopted ten modern CEC benchmark test functions, as provided in [49], for the pre-assessment of the proposed hybrid COAGWO algorithm. The details of the 10-digit challenge for single-objective optimization are listed in Table 2. Our aim here is to determine ten correct digits for each of the ten functions, ensuring that all functions achieve a minimum accuracy of 1.000000000. To demonstrate the effectiveness of the proposed COAGWO algorithm, we employed the Aquila-Grey Wolf Optimizer (AGWO) [46], the Reptile Search Algorithm (RSA) [50], the Whale Optimization Algorithm (WOA) [51], the Tunicate Swarm Algorithm (TSA) [52], and the Seagull Optimization Algorithm (SOA) [53]. We selected these algorithms for three reasons: (i) They consistently deliver remarkable results, proving effective in both benchmark and real-world problems. (ii) These algorithms are well cited and/or recently published in the literature. (iii) Implementations of these algorithms are accessible. The algorithms retained their original parameter settings without any modifications for problem-solving.

For the purpose of comparison, a population size of 30 was selected, along with 500 iterations. We run each algorithm 30 times. The average convergence rates of the algorithms are shown in Fig. 3. For all functions except for functions F01 and F06, the proposed COAGWO has the fastest convergence rate than competitor algorithms. Fig. 4 gives box-plots to show the data distributions of the 30 runs. A narrow boxplot shows that data has a high agreement. The boxplot of the proposed COAGWO is narrower than other algorithms for the majority of functions with the lowest values. Overall, the COAGWO algorithm performs better than other algorithms. Statistical results are presented in Table 3. The algorithm efficiency will be determined utilizing metrics of mean, standard deviation (Std), minimum (best), maximum (worst) and Mean Absolute Error (MAE) values for 30 runs. Their mathematical formulations are given as follows:

$$\text{mean} = \frac{1}{E_n} \sum_{i=1}^{E_n} f_{opt,i} \tag{20}$$

$$\text{Std} = \sqrt{\frac{1}{E_n - 1} \sum_{i=1}^{E_n} (f_{opt,i} - \text{mean})^2} \tag{21}$$

$$\text{minimum} = \min_{i=1}^{E_n} \{f_{opt,i}\} \tag{22}$$

$$\text{maximum} = \max_{i=1}^{E_n} \{f_{opt,i}\} \tag{23}$$

$$\text{MAE} = \frac{1}{E_n} \sum_{i=1}^{E_n} |f_{act,i} - f_{opt,i}| \tag{24}$$

where E_n is the number of experiments, $f_{act,i}$ represent the actual value and $f_{opt,i}$ is the optimal value in each independent run. As can be

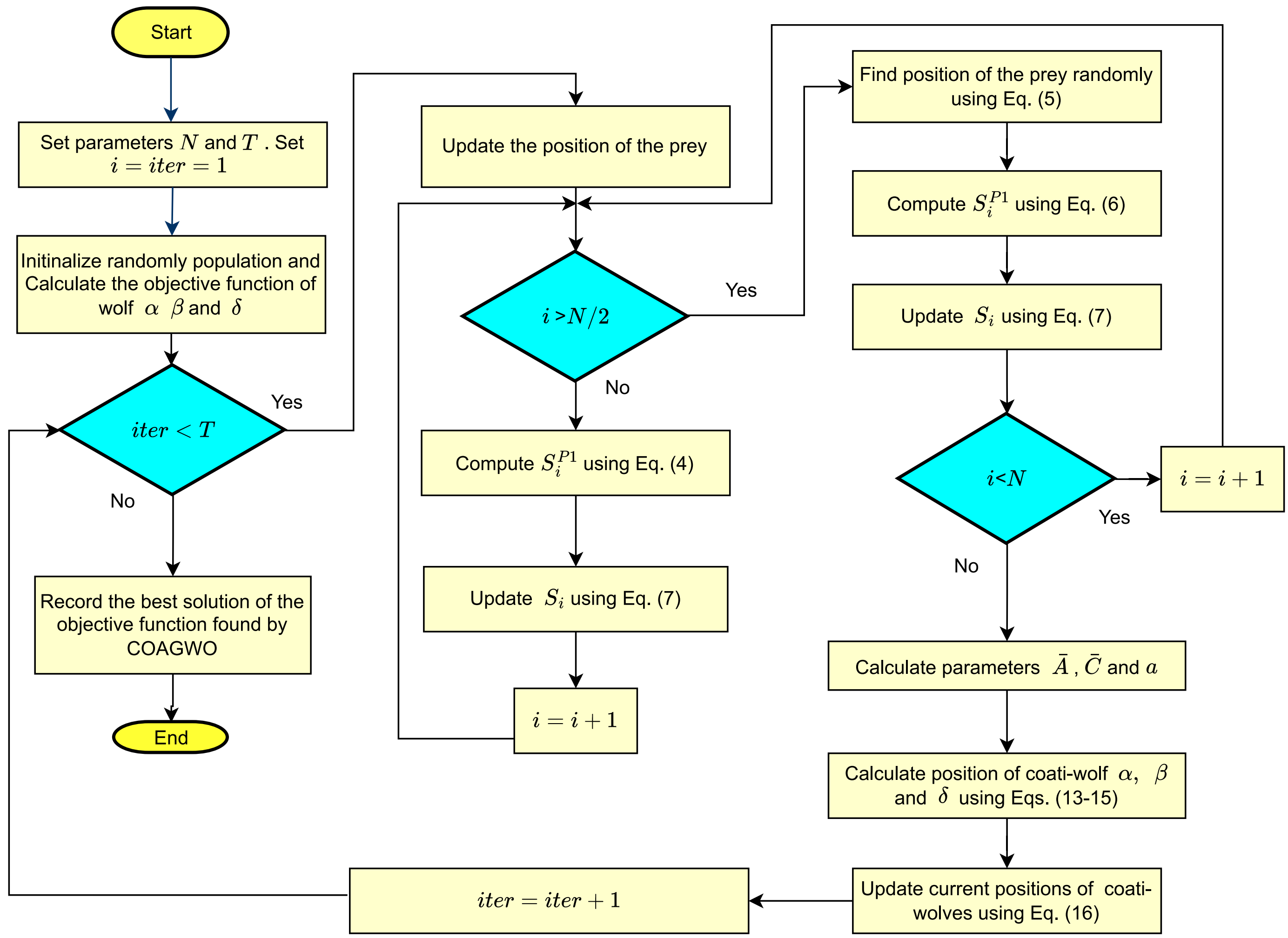


Fig. 2. Flowchart of the proposed COAGWO.

seen from the results presented in Table 3, our COAGWO algorithm outperforms the other seven algorithms on eight out of ten functions. Consequently, the COAGWO algorithm is ranked first for the overall performance of benchmark test problems compared to competitors. The findings demonstrate a well-balanced interplay between the exploration and exploitation phases within the proposed algorithm. Moreover, Table 4 gives the Wilcoxon signed-rank test results to determine whether there is a statistically significant difference in performance between the proposed algorithm and competitors at a significance level of 5% confidence, where symbol + indicates that the COAGWO algorithm significantly outperforms another algorithm while symbol − means the contrary and symbol = represents no significant difference. The performance results for COAGWO and the other algorithms are given in Table 4 along with a metric called the overall effectiveness (OE). We calculated OE as follows:

$$OE = ((T_N - T_L)/T_N) \times 100 \tag{25}$$

where T_N denotes the overall number of tests and T_L is the total number of unsuccessful tests for each algorithm. The COAGWO algorithm demonstrates a 70% OE when compared against GWO, COA, TSA, and SOA, and even higher OE percentages surpassing those of GWO, AGWO, RSA, and WOA.

In Table 5, an analysis of variance (ANOVA) is provided to assess whether there is a statistically significant difference between the means of the data obtained by the algorithms. The results of the ANOVA test show p-values lower than the significance level of 0.05 for all functions except function F10. For functions from F01 to F09, we can reject the null hypothesis and conclude that there are significant differences between the means of the data found by the algorithms.

4. Application of COAGWO to tuning optimal LQR control for an active suspension system

4.1. Model of a quarter-active suspension system

This section presents the equations of motion for a quarter-active suspension system, as depicted in Fig. 5. The active suspension system has two inputs: the control input F and the road surface position, z_r. Vehicle body displacement and tire displacement from the ground are denoted by z_s and z_{us}, respectively. The motion equations of the quarter-active suspension system are derived in [54] using Newton's laws as follows:

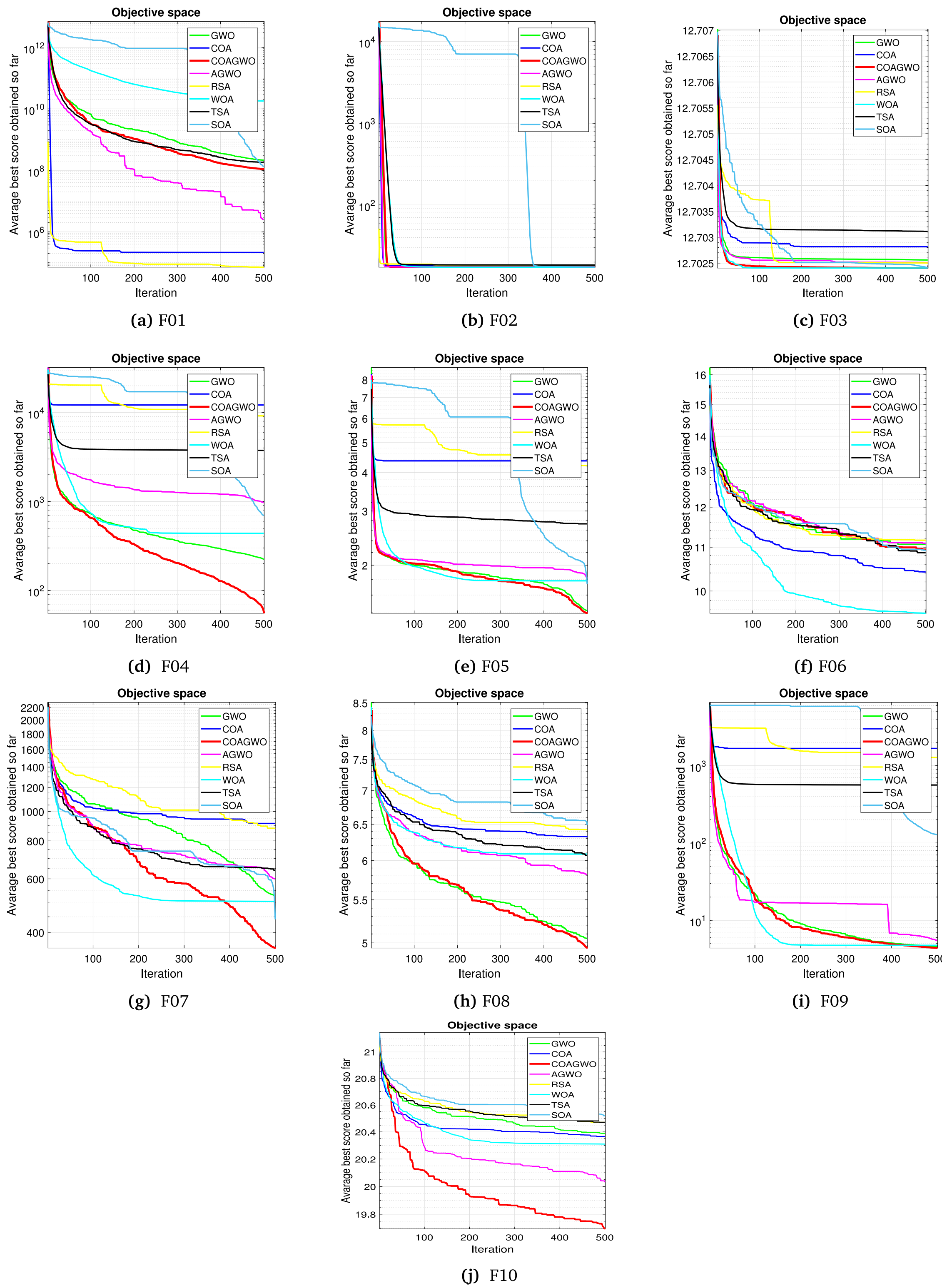
$$\begin{aligned} m_{us}\ddot{z}_{us} = {} & -b_{us}\dot{z}_{us} - F + b_s\dot{z} + b_{us}\dot{z}_r - (z_{us} - z_s)k_s \\ & - (z_{us} - z_r)k_{us} \end{aligned} \tag{26}$$

$$m_s\ddot{z}_s = b_s\dot{z}_{us} + F - b_s\dot{z}_s - (z_s - z_{us})k_s \tag{27}$$

Eqs. (26)–(27) can be given in the state–space realization as:

$$\begin{aligned} \dot{x}(t) &= \Phi x(t) + \Gamma u(t) \\ y(t) &= Lx(t) + Mu(t) \end{aligned} \tag{28}$$

where the state variable vector is $x(t) = [(z_s - z_{us}) \;\; \dot{z}_s \;\; (z_{us} - z_r) \;\; \dot{z}_{us}]'$ and the input vector is $u = [\dot{z}_r \;\; F]'$ and the output vector is $y = [(z_s - z_{us}) \;\; \ddot{z}_s]'$. $(z_s - z_{us})$ and $(z_{us} - z_r)$ are the suspension and tire



(a) F01 (b) F02 (c) F03 (d) F04 (e) F05 (f) F06 (g) F07 (h) F08 (i) F09 (j) F10

Fig. 3. Comparative study of COAGWO and competitor algorithms in objective place for each function.

deflections, respectively. $\dot{z}_s$ and $\dot{z}_{us}$ are the body and the tire vertical velocities respectively. Matrices Φ, Γ, L and M are obtained as follows:

$$\Phi = \begin{bmatrix} 0 & 1 & 0 & -1 \\ \frac{-k_s}{m_s} & \frac{-b_s}{m_s} & 0 & \frac{b_s}{m_s} \\ 0 & 0 & 0 & 1 \\ \frac{k_s}{m_{us}} & \frac{-b_s}{m_{us}} & \frac{k_{us}}{m_{us}} & -\frac{b_s+b_{us}}{m_{us}} \end{bmatrix}, \Gamma = \begin{bmatrix} 0 & 1 \\ 0 & \frac{1}{m_s} \\ -1 & 0 \\ \frac{b_{us}}{m_{us}} & -\frac{1}{m_{us}} \end{bmatrix} \tag{29}$$

$$L = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -\frac{k_s}{m_s} & -\frac{b_s}{m_s} & 0 & \frac{b_s}{m_s} \end{bmatrix}, M = \begin{bmatrix} 0 & 0 \\ 0 & \frac{1}{m_s} \end{bmatrix} \tag{30}$$

The specifications of the quarter-active suspension system are detailed in Table 6.

Table 3
Results of the experiments conducted with CEC2019 benchmark problems.

Function	Index	COAGWO (Proposed)	GWO	COA	AGWO	RSA	WOA	TSA	SOA
F01	Mean	1.0984e+08	2.1508e+08	2.1390e+05	2.5439e+06	7.1201e+04	1.8060e+10	1.8123e+08	1.3194e+08
	Std	2.2377e+08	3.1400e+08	1.9006e+05	1.3933e+07	2.0090e+04	2.7080e+10	3.3367e+08	3.4803e+08
	Best	3.5861e+05	1.2202e+06	4.3069e+04	0	4.6843e+04	7.4318e+05	4.5342e+04	4.3124e+04
	Worst	1.1970e+09	1.2425e+09	7.7804e+05	7.6316e+07	1.2801e+05	1.1376e+11	1.2119e+09	1.8434e+09
	MAE	1.0976e+08	2.1508e+08	2.1390e+05	2.5439e+06	7.1200e+04	1.8060e+10	1.8123e+08	1.3194e+08
	Rank	4	7	2	3	1	8	6	5
F02	Mean	17.3439	17.3664	18.0322	17.3889	18.2374	17.3521	18.4308	17.4232
	Std	3.2068e−04	0.0860	0.3031	0.1100	0.5095	0.0098	0.6783	0.1249
	Best	17.3432	17.3434	17.5529	17.3433	17.6992	17.3442	17.3535	17.3470
	Worst	17.3446	17.6876	18.5514	17.6772	19.5122	17.3929	19.5649	17.6841
	MAE	16.3435	16.3664	17.0322	16.3889	17.2374	16.3521	17.4308	16.4232
	Rank	1	3	6	4	7	2	8	5
F03	Mean	12.7024043	12.7025599	12.7028158	12.7025106	12.7025023	12.7024046	12.7031139	12.7024094
	Std	4.7889e−07	5.8476e−04	6.2503e−04	3.0050e−04	1.7346e−05	9.0524e−07	0.0013	3.1499e−06
	Best	12.7024042	12.7024042	12.7024051	12.7024042	12.702435	12.7024042	12.7024043	12.7024046
	Worst	12.7024	12.7049	12.7047	12.7034	12.7025	12.7024	12.7074	12.7024
	MAE	11.7024	11.7026	11.7028	11.7025	11.7025	11.7024	11.7031	11.7024
	Rank	1	6	7	5	4	2	8	3
F04	Mean	57.5189	222.0273	1.2186e+04	972.6407	9.1744e+03	439.1962	3.7571e+03	698.0885
	Std	21.6930	609.0832	5.4922e+03	1.2922e+03	2.2685e+03	423.2868	2.3654e+03	793.1093
	Best	26.3132	17.4823	2.9893e+03	79.0734	6.3451e+03	144.4515	135.5076	67.0229
	Worst	120.7365	2.4656e+03	2.2317e+04	3.8626e+03	1.7592e+04	2.4326e+03	8.5779e+03	3.3022e+03
	MAE	54.3913	221.0273	1.2185e+04	971.6407	9.1734e+03	438.1962	3.7561e+03	697.0885
	Rank	1	2	8	5	7	3	6	4
F05	Mean	1.3977	1.4248	4.3630	1.8149	4.2155	1.7799	2.7248	1.7389
	Std	0.2168	0.2772	0.8097	0.1914	0.5857	0.3646	0.6584	0.2703
	Best	1.0847	1.0414	3.0672	1.3438	3.1221	1.2449	1.4164	1.3651
	Worst	1.7968	1.8952	6.3491	2.1161	5.5132	3.2255	4.5278	2.5407
	MAE	0.3969	0.4248	3.3630	0.8149	3.2155	0.7799	1.7248	0.7389
	Rank	1	2	8	5	7	4	6	3
F06	Mean	10.9448	11.0688	10.4253	11.0950	11.1685	9.5298	10.8679	10.9144
	Std	0.6796	0.8021	0.7792	0.6533	0.6147	1.3330	0.7136	0.7117
	Best	8.8533	9.0657	8.9162	9.2759	9.7080	6.7428	8.5134	8.4452
	Worst	11.8339	12.1971	11.9513	12.5988	11.8879	11.4446	11.9115	11.8588
	MAE	9.9448	10.0688	9.4253	10.0950	10.1685	8.5298	9.8679	9.9144
	Rank	5	6	2	7	8	1	3	4
F07	Mean	355.4440	528.2632	913.1503	600.0505	877.9409	505.9453	640.5778	441.3233
	Std	281.0799	290.3691	224.1709	279.6131	177.8949	266.9320	227.6688	168.9149
	Best	−3.2986	80.5383	545.4950	77.5738	356.8096	125.3847	245.8240	97.3229
	Worst	911.2226	1.1315e+03	1.3259e+03	1.1144e+03	1.1882e+03	1.2534e+03	1.1042e+03	900.3634
	MAE	354.5625	527.2632	912.1503	599.0505	876.9409	504.9453	639.5778	440.3233
	Rank	1	4	8	5	7	3	6	2
F08	Mean	4.9450	5.0498	6.3167	5.8161	6.4224	6.0845	6.0665	6.4884
	Std	0.9539	1.0319	0.3409	0.5732	0.3163	0.5023	0.6993	0.3954
	Best	3.0845	1.7717	5.6064	4.4803	5.7441	4.2745	4.2368	5.8468
	Worst	6.7678	6.9764	6.9169	6.9611	6.9106	6.7032	7.0798	7.2691
	MAE	3.9441	4.0498	5.3167	4.8161	5.4224	5.0845	5.0665	5.4884
	Rank	1	2	6	3	7	5	4	8
F09	Mean	4.3852	4.5833	1.6716e+03	5.4866	1.2785e+03	4.7489	562.5075	129.0596
	Std	1.2472	0.7984	581.5690	0.5555	421.2172	1.0152	818.9090	328.3620
	Best	3.0730	3.1609	646.1030	4.0181	361.0108	2.8539	3.4774	2.7897
	Worst	9.3916	6.3244	2.9615e+03	6.1865	2.2540e+03	6.4781	3.6928e+03	1.7198e+03
	MAE	3.3835	3.5833	1.6706e+03	4.4866	1.2775e+03	3.7489	561.5075	128.0596
	Rank	1	2	8	4	7	3	6	5
F10	Mean	19.6960	20.3895	20.3660	20.0315	20.4711	20.3098	20.4712	20.5199
	Std	3.0172	0.5363	0.1320	1.7547	0.1387	0.1293	0.1029	0.0972
	Best	6.3286	17.5925	20.1015	11.8053	19.9139	20.1068	20.1908	20.3085
	Worst	20.6559	20.6142	20.6286	20.6766	20.6549	20.5647	20.6222	20.6581
	MAE	18.6948	19.3895	19.3660	19.0315	19.4711	19.3098	19.4712	19.5199
	Rank	1	5	4	2	6	3	7	8
Sum rank		19	39	53	43	61	34	62	47
Mean rank		1.9	3.9	5.3	4.3	6.1	3.4	6.2	4.7
Total rank		1	3	6	4	7	2	8	5

4.2. Road input model

Vibration occurs when a car drives on rough roads. A controller is developed to minimize body acceleration within an active suspension system. The road input models will be introduced to examine the efficacy of the active suspension system. Three kinds of road disturbances are included in the analysis. The first one is a square signal with a magnitude of 0.01 m and a frequency of 0.3 Hz. The second road profile

Table 4
Statistical results of the Wilcoxon rank-sum test with CEC2019 benchmark problems.

COAGWO (proposed) vs.	GWO		COA		AGWO		RSA		WOA		TSA		SOA	
Function	p-value	Win	p-value	Win	p-value	Win	p-value	Win	p-value	Win	p-value	Win	p-value	Win
F01	3.0199e−11	+	7.3891e−11	−	1.2620e−11	−	3.0199e−11	−	1.1937e−06	+	0.2226	=	0.0748	=
F02	0.8650	=	3.0199e−11	+	1.2870e−09	+	3.0199e−11	+	8.1527e−11	+	3.0199e−11	+	3.0199e−11	+
F03	0.8766	=	3.3176e−11	+	2.0152e−08	+	3.0199e−11	+	0.0019	+	4.1997e−10	+	8.1527e−11	+
F04	0.2772	=	3.0199e−11	+	3.4742e−10	+	3.0199e−11	+	3.0199e−11	+	3.0199e−11	+	1.9568e−10	+
F05	8.303e−06	+	3.0199e−11	+	3.3520e−08	+	3.0199e−11	+	8.2919e−06	+	1.2057e−10	+	1.0188e−05	+
F06	0.2643	=	0.0061	−	0.7283	=	0.1453	=	1.1674e−05	−	0.5592	=	0.7394	=
F07	0.0163	+	1.3111e−08	+	0.0018	+	1.8500e−08	+	0.0315	+	1.2477e−04	+	0.0679	=
F08	0.6735	=	3.9648e−08	+	1.0407e−04	+	1.1023e−08	+	1.6062e−06	+	4.4205e−06	+	7.1186e−09	+
F09	1.669e−08	+	3.0199e−11	+	1.6062e−06	+	3.0199e−11	+	0.0701	=	1.2023e−08	+	3.0811e−08	+
F10	0.3871	=	0.0207	−	0.9587	=	0.4204	=	6.5486e−04	−	0.5692	=	0.0378	+
Total +/ − / =	4/0/5		7/3/0		7/1/2		7/1/2		7/2/1		7/0/3		7/0/3	
OE	%100		%70		%90		%90		%80		%70		%70	

is a bumpy signal, which is given as:

$$z_r(t) = \begin{cases} \frac{0.01}{2}(1 - cos5\pi t) & 0 \le t \le 0.4 \\ 0 & \text{otherwise} \end{cases} \tag{31}$$

Lastly, many driving conditions can be characterized as a random road input model. Therefore, the class C-road profile [55] is employed, which is given as follows:

$$\dot{z}_r(t) = -2\pi f_0 z_r(t) + 2\pi n_0 w(t)\sqrt{G_0 v} \tag{32}$$

where G_0 is the road roughness coefficient, and for the standard C-grade surface, $G_0 = 256 \times 10^{-6}$ m^3; $w(t)$ is the random white noise signal; the lower cut-off frequency f_0 is 0.0628 Hz; the velocity of the vehicle is $v = 30$ m/s; the frequency index is $n_0 = 0.1$ m^{-1}.

4.3. Control objectives

An optimal LQR controller will be developed to ensure the following performance requirements given in [8].

- Ride comfort of passengers: The vehicle body acceleration, $\ddot{z}_s$ must be reduced by the active suspension system.
- Handling stability: The active suspension has to maintain the suspension deflection within the allowable interval to avoid vehicle damage. $|(z_s - z_{us})| \le \bar{z}$, where $\bar{z}$ is the greatest acceptable suspension deflection.
- Driving safety: The wheel assembly has to stay in firm contact with the road to ensure passenger safety. Therefore, the tire's dynamic load has to be smaller than its static load $(k_s(z_{us} - z_r) \le (m_s + m_{us})g)$.

4.4. Problem formulation

Consider the linear time-invariant system given in (28) and the purpose is to find the optimal control law, $u(t)$ which can drive the state variables of the dynamics to demand state by optimizing the following quadratic objective function:

$$J = \int_0^T \left(x(t)'\mathcal{Q}x(t) + u'(t)\mathcal{R}u(t) \right) dt \tag{33}$$

Here $\mathcal{Q}$ is the positive semi-definite state and $\mathcal{R}$ is the positive-definite weighting matrices respectively. Diagonal weighting matrices are generally selected. The order of Q and R matrices equals the number of states and the number of inputs. Assume that (Φ, Γ) is stabilizable and (Φ, L) is observable, then the LQR controller computes as follows:

$$u(t) = -\mathcal{K}x(t) \tag{34}$$

in which $\mathcal{K}$ is the optimal state-feedback gain computed by $\mathcal{K} = R^{-1}\mathcal{B}'\mathcal{P}$ that is called the Lagrange multiplier based on optimization. The positive definite-matrix, $\mathcal{P}$ is obtained from the solution of the following algebraic Riccati equation:

$$\Phi'\mathcal{P} + \mathcal{P}\Phi + \mathcal{Q} - \mathcal{P}\Gamma\mathcal{R}^{-1}\Gamma'\mathcal{P} = 0 \tag{35}$$

The design of the LQR control has a challenging issue in selecting the weighting matrices. Opting for specific weighting matrices $\mathcal{Q}$ and $\mathcal{R}$ can influence the rate of change in state variables and control effort. Hence, the weighting matrices can be optimized by using the proposed COAGWO algorithm in this section. The model of the augmented system is given as a fourth-order system. Therefore, the weighting matrix $\mathcal{Q}$ is set to be a 4×4 diagonal semi-definite matrix ($\mathcal{Q} = diag([q_1\ q_2\ q_3\ q_4])$. The system has a control input F, thereby $\mathcal{R}$ is considered as a scalar $(\mathcal{R} = v_1)$. The corresponding J to be minimized is given as follows:

$$J = \int_0^\infty \left(q_1 x_1^2 + q_2 x_2^2 + q_3 x_3^2 + q_4 x_4^2 + v_1 u_1^2 \right) dt \tag{36}$$

in which q_1, q_2, q_3 and q_4 are scalar weights of state variables. r_1 is the scalar weight of the controlled force. LQR weighting matrices are optimized using the fitness function that is given by the following integral of the square error (ISE):

$$ISE = \int_0^\infty 0.2e_{x_1}^2(t) + 0.6e_{x_2}^2(t) + 0.1e_{x_3}^2(t) + 0.1e_{x_4}^2(t)dt \tag{37}$$

where $e_{x_1}(t)$, $e_{x_2}(t)$, $e_{x_3}(t)$, and $e_{x_4}(t)$ are tracking errors of state variables. The parameters of the algorithms and bound of each element of matrices $\mathcal{Q}$ and $\mathcal{R}$ are given in Table 7.

Fig. 6 displays the comparative objective function (ISE) values with respect to the number of runs for the controlled active suspension system under the square road disturbance. As seen in this plot, the proposed COAGWO performs better results in 30 out of 30 runs compared to GWO and COA algorithms. In Fig. 7, we compare the average convergence curves of the proposed COAGWO, GWO and COA algorithms. As seen from the figure, the convergence performance of our algorithm is better than others. Fig. 8 shows boxplot results for each algorithm. The proposed COAGWO has more robust stability than other algorithms in objective function, ISE. Fig. 9 shows exploration and exploitation ratios during the search process maintained by different algorithms while solving optimal LQR control of the suspension system. The COAGWO algorithm exhibits a balanced approach between exploration and exploitation, as evidenced by the bar charts (Fig. 10) depicting the average exploration and exploitation percentages of the algorithms Table 8 gives comparative statistical results of objective function values. The proposed COAGWO obtains the lowest objective function value and provides the lowest values for mean, minimum (Best) and maximum (Worst). Table 9 shows the p-values of the proposed COAGWO against compared algorithms for the active suspension system, while Table 10 presents the ANOVA test for GWO, COA and COAGWO for the active suspension system. The results of these tables confirm the superiority of the proposed COAGWO algorithm and hence our algorithm is ranked

Table 5
One-way analysis of variance (ANOVA) test for CEC2019 benchmark problems.

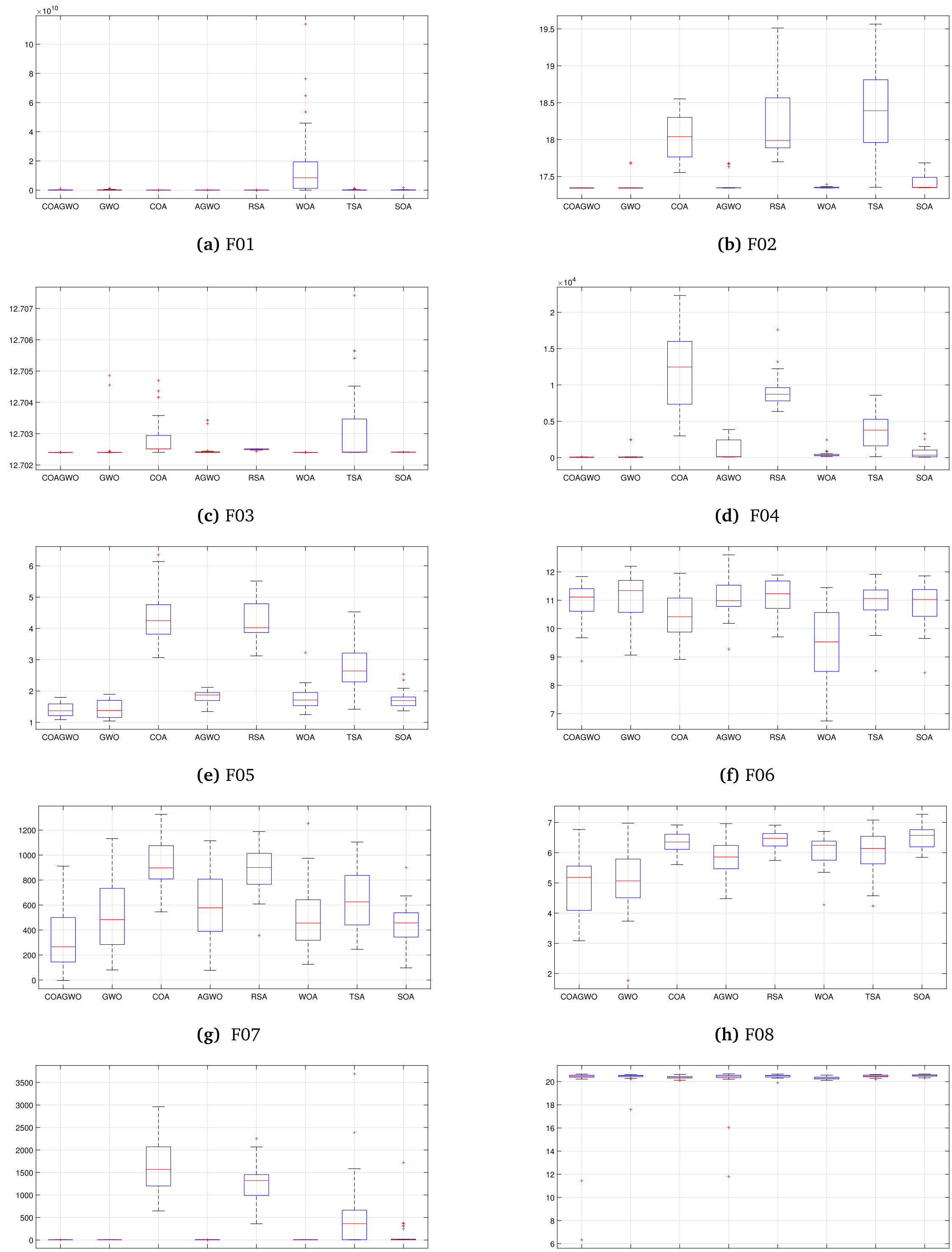
F01					
	SS	DF	MS	F_a	P-value
Between algorithms	8.47688e+21	7	1.21098e+21	13.2	2.64718e−14
Within algorithms	2.12774e+22	232	9.17131e+19		
Total	2.97543e+22	239			
F02					
	SS	DF	MS	F_a	P-value
Between algorithms	43.9699	7	6.28142	59.35	3.00497e−48
Within algorithms	24.5556	232	0.10584		
Total	68.5255	239			
F03					
	SS	DF	MS	F_a	P-value
Between algorithms	1.32554e−05	7	1.89363e−06	6.02	1.82648e−06
Within algorithms	7.29307e−05	232	3.14356e−07		
Total	8.61861e−05	239			
F04					
	SS	DF	MS	F_a	P-value
Between algorithms	4.61626e+09	7	6.59466e+08	120.57	1.26073e−73
Within algorithms	1.26891e+09	232	5.46945e+06		
Total	5.88518e+09	239			
F05					
	SS	DF	MS	F_a	P-value
Between algorithms	310.979	7	44.4255	197.59	5.40419e−94
Within algorithms	52.161	232	0.2248		
Total	363.14	239			
F06					
	SS	DF	MS	F_a	P-value
Between algorithms	62.069	7	8.86698	13.36	1.82877e−14
Within algorithms	153.975	232	0.66369		
Total	216.044	239			
F07					
	SS	DF	MS	F_a	P-value
Between algorithms	8.26344e+06	7	1180490.9	19.88	8.56517e−21
Within algorithms	1.37756e+07	232	59 377.6		
Total	2.2039e+07	239			
F08					
	SS	DF	MS	F_a	P-value
Between algorithms	74.892	7	10.6988	25.05	2.48811e−25
Within algorithms	99.099	232	0.4272		
Total	173.991	239			
F09					
	SS	DF	MS	F_a	P-value
Between algorithms	9.26018e+07	7	13228828.4	81.78	4.44738e−59
Within algorithms	3.75284e+07	232	161 760.4		
Total	1.3013e+08	239			
F10					
	SS	DF	MS	F_a	P-value
Between algorithms	16.611	7	2.37295	1.51	0.1635
Within algorithms	363.773	232	1.56799		
Total	380.383	239			

Table 6
Model parameters.

Symbol	Value	Definition
m_s	2.45 kg	Main body mass
m_{us}	1 kg	Wheel assembly mass
k_s	900 N/m	Chassis stiffness
k_{us}	1250 N/m	Suspension stiffness
b_s	7.5 N s/m	Chassis damping coefficient
b_{us}	5 N s/m	Tire inherent damping coefficient
$\bar{z}$	0.038 m	Maximum suspension deflection

Table 7
The parameters of the proposed COAGWO, GWO and COA algorithms for designing LQR controllers.

Parameters	COAGWO	GWO	COA
Population size	20	20	20
Maximum iteration number	50	50	50
Run number	30	30	30
Variable number	5	5	5
Lower bound of $[q_1\ q_2\ q_3\ q_4\ r_1]$	lb = 10^{-2}[1 1 1 1 1]	lb	lb
Upper bound of $[q_1\ q_2\ q_3\ q_4\ r_1]$	ub = 10^{2}[5 5 5 5 5 5]	ub	ub



(a) F01 **(b)** F02 **(c)** F03 **(d)** F04 **(e)** F05 **(f)** F06 **(g)** F07 **(h)** F08 **(i)** F09 **(j)** F10

Fig. 4. Boxplot of COAGWO and competitor algorithms in the optimization of the CEC2019 benchmark problems.

first compared to other algorithms Table 11 gives the optimal state and input weighting matrices of the LQR controller obtained using COAGWO, GWO and COA algorithms.

5. Simulation results

This section presents the detailed simulation results to verify the efficacy of the proposed COAGWO tune-LQR controller under different road disturbance profiles. All simulations for solving optimal weighting matrices are implemented using MATLAB\Simulink on a Windows 10 Home intel^{R} Core™ i5-3320M CPU at 2.60 GHZ, 8 GB RAM. The simulation results under the different road profiles are compared in Fig. 11. The peak-to-peak (PTP), root mean square (RMS) and improvement in RMS values of the suspension system are reported in Table 12. Compared with the passive system, the active suspension system with the proposed COAGWO algorithm has significantly reduced

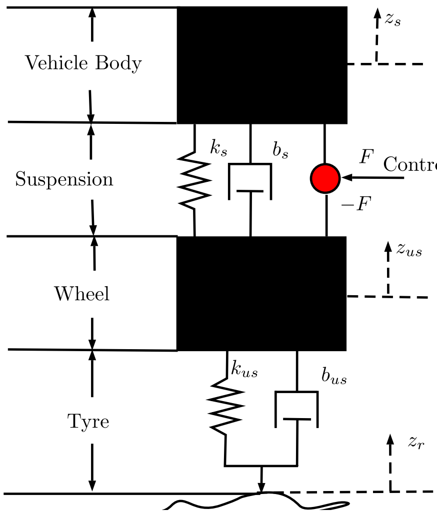


Fig. 5. A quarter active suspension system.

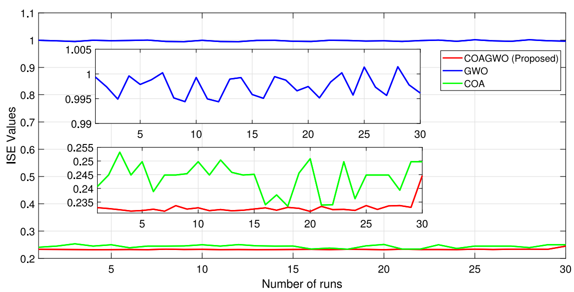


Fig. 6. Obtained ISE values from 30 runs for the proposed COAGWO, GWO and COA algorithms.

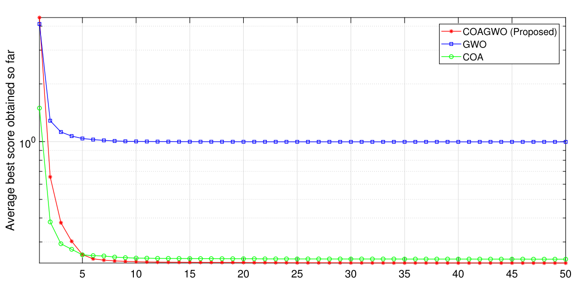


Fig. 7. The average evolving process of ISE values for the proposed COAGWO, GWO and COA algorithms.

Fig. 8. The boxplot of the proposed COAGWO, GWO and COA algorithms.

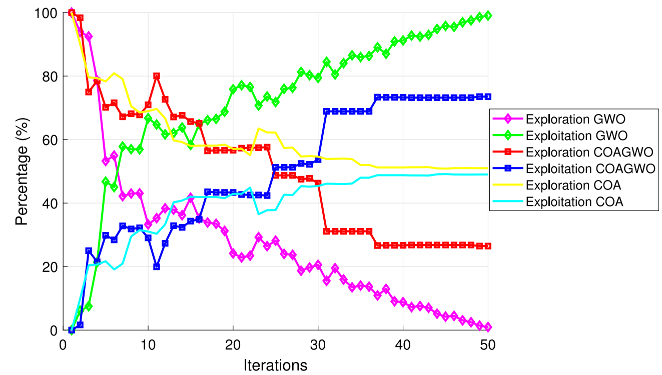


Fig. 9. Exploration and exploitation results of algorithms for obtained ISE values.

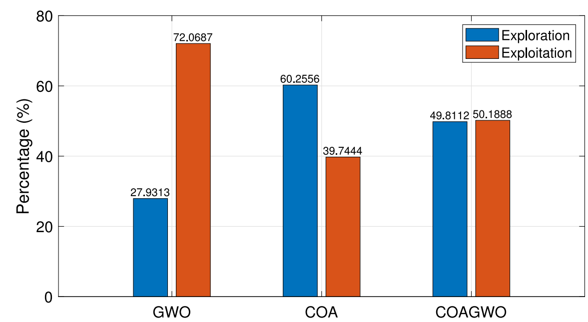


Fig. 10. Average exploration and exploitation of algorithms for obtained ISE values.

Table 8
Comparison of statistical results for obtained ISE values.

Index	GWO	COA	COAGWO
Mean	0.9976	0.2439	0.2329
std	0.0021	0.0056	0.0023
Best	0.9944	0.2335	0.2316
Worst	1.0015	0.2532	0.2446
MAE	0.9976	0.2439	0.2329
Rank	3	2	1

Table 9
Results of Wilcoxon rank-sum test for obtained ISE values.

Wilcoxon rank-sum test				
COAGWO (proposed) vs.	GWO		COA	
	P-value	Win	P-value	Win
	3.0199e−11	+	9.5271e−09	+
OE	%100		%100	

Table 10
Results of one-way analysis of variance (ANOVA) test for obtained ISE values.

Source of variation	SS	DF	MS	F_a	P-value
Between algorithms	11.5116	2	5.75581	359 394.77	4.02369e−171
Within algorithms	0.0014	87	0.00002		
Total	11.513	89			

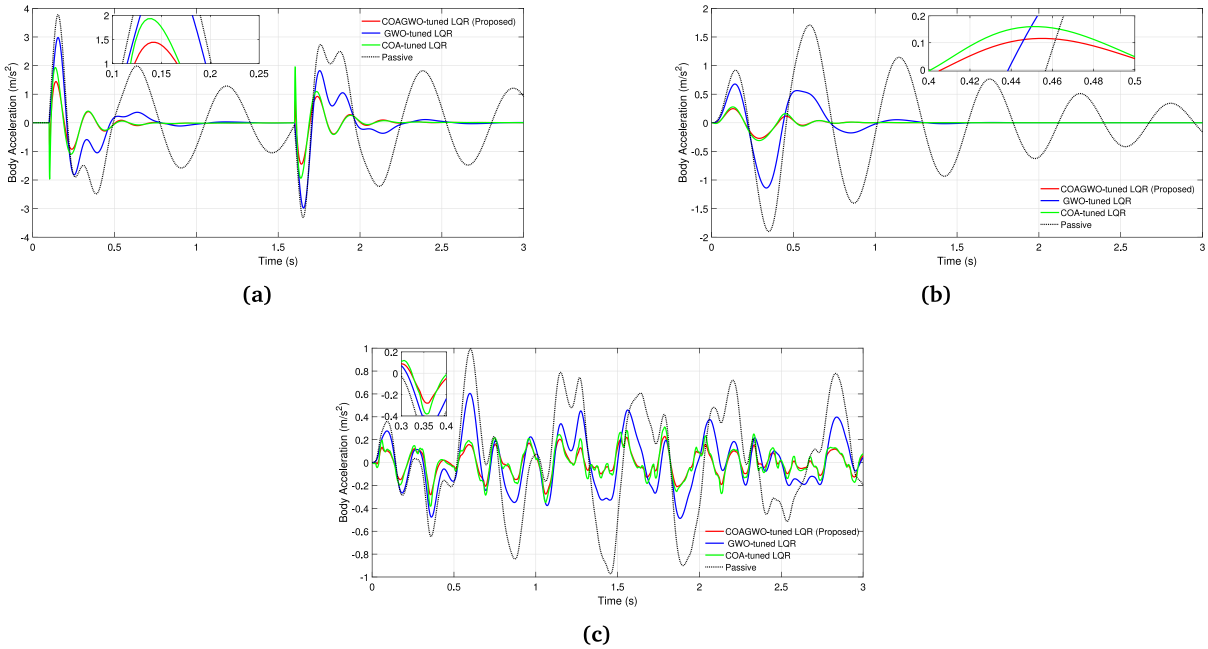


Fig. 11. Body acceleration responses of passive and the active suspension system with LQR controllers based on COAGWO, GWO and COA algorithms under (a) square road, (b) bumpy road, and (c) class-C road disturbances.

Table 11
Obtained weight matrices using different algorithms for LQR controllers.

COAGWO	$Q = diag([2.8253, \quad 493.6680, \quad 361.5294, \quad 7.7928])$ $\mathcal{R} = 0.01$
GWO	$Q = diag([2.5990, \quad 10, \quad 10 \quad 0.01])$ $\mathcal{R} = 0.01$
COA	$Q = diag([0.02, \quad 10.8696, \quad 0.02 \quad 0.3358])$ $\mathcal{R} = 0.021739$

the RMS values of the vehicle body acceleration. For the square, bumpy and C-grade road profiles, the RMS values of body acceleration are reduced by 77.57%, 90.06% and 77.40 respectively. One can see that from Table 12, the GWO tuned-LQR controller has obtained the lowest improvement in RMS values.

5.1. Comparison of the COAGWO-tuned LQR with the published works

The proposed COAGWO-tuned LQR controller is also compared with the published works which obtained optimal LQR controllers for the active suspension system. The available and promising approaches are cited as GWO-tuned LQR [13], BA-tuned LQR [13], ACO-tuned LQR [7], AIWPSO-tuned LQR [8], and APPO-tuned LQR [15]. The comparative analysis of controllers in terms of the vehicle body acceleration, vehicle body position, suspension deflection and tire deflection under bumpy road profile is provided in Table 13. As can be seen from Fig. 12, the proposed COAGWO-tuned controller has a clear superiority in terms of vehicle body acceleration, body position and tire dynamic load responses.

Likewise, a comparative performance evaluation is also performed in the case of the standard C-grade road profile in terms of the vehicle body acceleration, vehicle body position, suspension deflection and tire deflection. Regarding simulation results given in Table 14, it can be seen that the maximum value of the suspension deflection for the proposed COAGWO tuned-LQR controller is 0.009686 m which is less than the permissible travel range of 0.038 m. For road handling requirements, the tire's dynamic load has to be smaller than its static load. All designed controllers obtain that tire dynamic loads are less than 33.84 N. Similar to the case of bumpy road disturbance, the proposed COAGWO is able to reduce vehicle body acceleration under the standard C-grade road profile as well (see Fig. 13). All comparative results verify that the COAGWO-tuned LQR controller satisfies the control objectives given in Section 4.3 and the proposed LQR controller is superior to other optimal LQR controllers in the recent literature.

6. Discussion

This section discusses the validation techniques applied to ensure the reliability of the results. Extensive experiments were conducted using the CEC 2019 benchmarks. The comparison in Table 3 provides a detailed assessment of various optimization algorithms across multiple benchmark functions, including COAGWO, AGWO [46], RSA [50], WOA [51], TSA [52] and SOA [53]. Each algorithm's performance is evaluated using statistical metrics such as mean, standard deviation, best, worst, and mean absolute error. COAGWO consistently demonstrates superior optimization results compared to other algorithms, achieving the lowest mean values in several functions. Its robust and consistent performance across various benchmark functions indicates its effectiveness. In contrast, algorithms like WOA and SOA exhibit higher variances and poorer performance, resulting in lower ranks. While COAGWO generally outperforms other algorithms, there are exceptional functions F06 and F10 where alternatives perform better. Wilcoxon test results in Table 4 show that COAGWO significantly outperforms other algorithms, with a 100% overall effectiveness against GWO, 90% against COA and AGWO, 80% against RSA, and 70% against WOA, TSA, and SOA. This indicates COAGWO's dominance in minimizing objective function values across various optimization tasks. ANOVA test results in Table 5 reveal statistically significant differences in algorithm performance for most benchmark functions (F01 to F09), suggesting that the choice of algorithm significantly impacts optimization outcomes. However, for function F10, there is no significant difference in performance among the algorithms, indicating similar performance levels. Overall, the comprehensive evaluation highlights the efficiency, reliability, and robustness of COAGWO in optimization problems.

For the optimal weight selection problem for linear quadratic control applied to vehicle suspension systems, the COAGWO algorithm

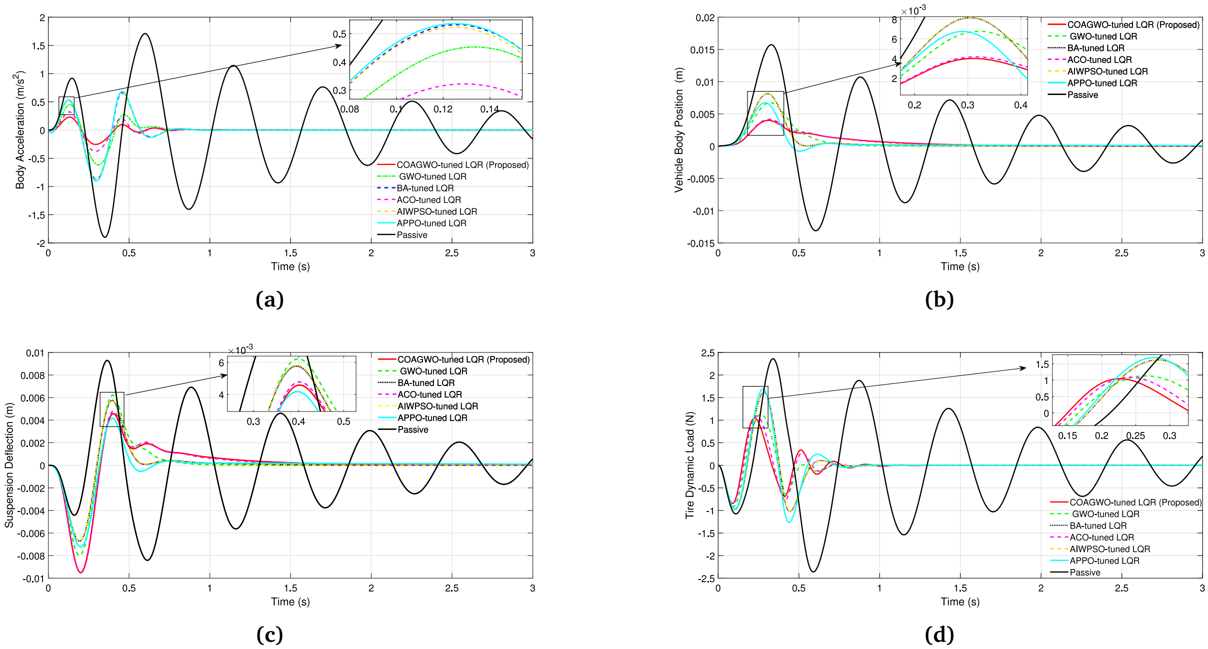


Fig. 12. (a) body acceleration, (b) vehicle body displacement, (c) suspension deflection, and (d) tire dynamic load responses of the passive and the active suspension system with different controllers under the bumpy road profile.

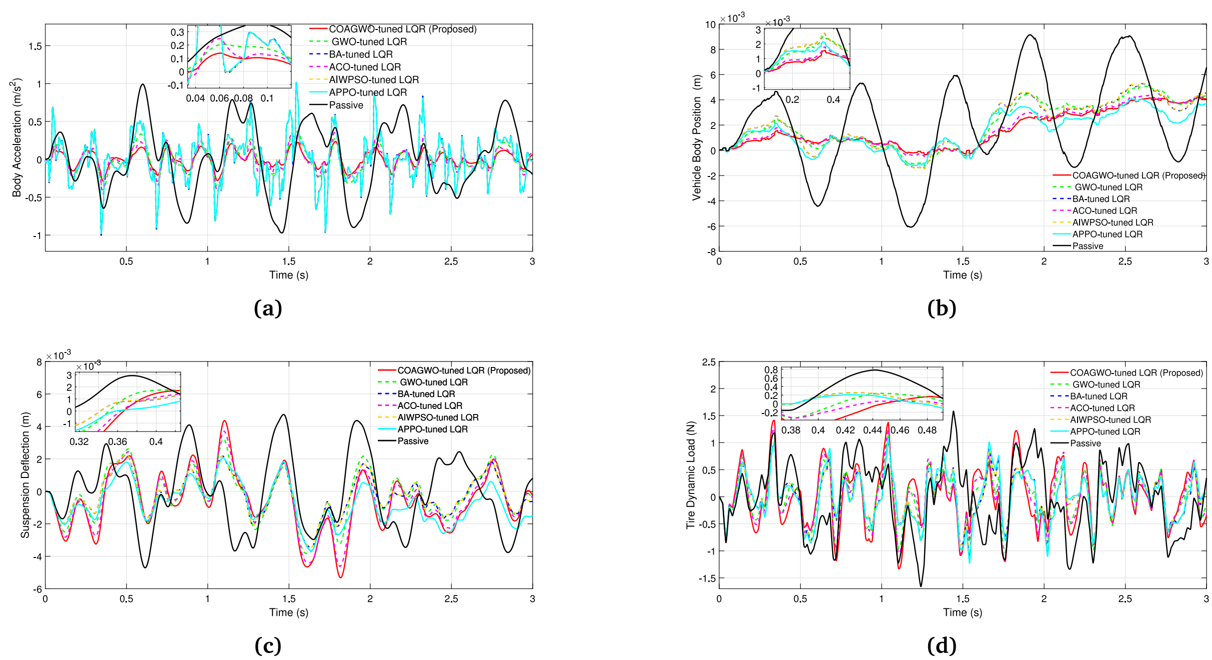


Fig. 13. (a) body acceleration, (b) vehicle body displacement, (c) suspension deflection, and (d) tire dynamic load responses of passive and the active suspension system with different controllers under the standard C-grade road surface.

surpasses both the GWO and COA algorithms across all evaluated metrics in Table 8. COAGWO exhibits the lowest mean, signifying superior overall performance, along with low variability, as demonstrated by its standard deviation comparable to GWO. Moreover, COAGWO achieves the best results underscoring its robustness and efficiency in minimizing the ISE value. These findings collectively place COAGWO as the top-performing algorithm. The Wilcoxon rank-sum test results in Table 9 further reinforce COAGWO's superiority over GWO and COA in minimizing the ISE value. The significant p-value shows COAGWO's effectiveness in leading to improved control performance. COAGWO exhibits a 100% overall effectiveness against both GWO and COA, indicating consistent superiority. The ANOVA results in Table 10 corroborate the significant differences in ISE values obtained from different algorithms. According to Tables 12–14 the comparison of COAGWO-LQR against other controllers in terms of RMS values of $\ddot{z}$, $z_s - z_{us}$, and $k_s(z_{us} - z_r)$ metrics consistently demonstrates the superiority of the COAGWO-LQR controller. The COAGWO-LQR controller achieves significant improvements ranging from 30.21% to 80.10% compared

Table 12
Performance comparison of LQR controllers based on the proposed COAGW, GWO and COA algorithm under different road profiles.

Controller	Square road profile			Bumpy road profile		
	PTP $\ddot{z}_s$ (m/s^2)	RMS $\ddot{z}_s$ (m/s^2)	Improvement in RMS $\ddot{z}_s$ (%)	PTP $\ddot{z}_s$ (m/s^2)	RMS $\ddot{z}_s$ (m/s^2)	Improvement in RMS $\ddot{z}_s$ (%)
COAGWO-LQR	2.887	0.3254	77.57	0.524	0.0761	90.06
GWO-LQR	5.963	0.7233	50.15	1.818	0.314	58.92
COA-LQR	3.923	0.4399	69.68	0.588	0.085	88.84
Passive	7.089	1.4510	–	3.60	0.766	–
Controller	C-grade road profile					
	PTP $\ddot{z}_s$ (m/s^2)	RMS $\ddot{z}_s$ (m/s^2)	Improvement in RMS $\ddot{z}_s$ (%)			
COAGWO-LQR	0.522	0.1041	77.40			
GWO-LQR	5.963	0.7233	50.15			
COA-LQR	0.693	0.1289	72.02			
Passive	1.964	0.4607	–			

Table 13
Performance assessment of the active suspension system employing different controllers under the bumpy road profile.

Controller	PTP $\ddot{z}_s$ (m/s^2)	RMS $\ddot{z}_s$ (m/s^2)	Improvement in RMS $\ddot{z}_s$ (%)	PTP $z_s - z_{us}$ (m)	RMS $z_s - z_{us}$ (m)	Improvement in RMS $z_s - z_{us}$ (%)
COAGWO-LQR	0.5242	0.07611	90.06	0.01448	0.002624	30.21
GWO-LQR [13]	1.069	0.1630	78.72	0.01421	0.002309	38.59
BA-LQR [13]	1.530	0.2656	65.33	0.01149	0.002101	44.12
ACO-LQR [7]	0.7002	0.1027	86.59	0.01425	0.002522	32.92
AIWPSO-LQR [8]	1.525	0.2599	66.07	0.01148	0.002179	42.04
APPO-LQR [15]	1.589	0.2809	63.33	0.01140	0.002145	42.95
Passive	3.609	0.7662	–	0.01771	0.00376	–
Controller	PTP $k_s(z_{us} - z_r)$ (N)	RMS $k_s(z_{us} - z_r)$ (N)	Improvement in RMS $k_s(z_{us} - z_r)$ (%)	PTP z_s (m)	RMS z_s (m)	Improvement in RMS z_s (%)
COAGWO-LQR	1.888	0.2826	71.85	0.003465	0.001246	80.10
GWO-LQR [13]	2.046	0.3087	69.25	0.006765	0.001988	68.25
BA-LQR [13]	2.844	0.49888	50.31	0.007300	0.001835	70.69
ACO-LQR [7]	1.936	0.2985	70.20	0.00416	0.001378	77.99
AIWPSO-LQR [8]	2.885	0.5021	49.99	0.007119	0.001798	71.28
APPO-LQR [15]	2.956	0.5348	46.73	0.007518	0.002017	67.78
Passive	4.721	1.004	–	0.2884	0.006262	–

Table 14
Performance assessment of the active suspension system employing different controllers under the standard C-grade road surface.

Controller	PTP $\ddot{z}_s$ (m/s^2)	RMS $\ddot{z}_s$ (m/s^2)	Improvement in RMS $\ddot{z}_s$ (%)	PTP $z_s - z_{us}$ (m)	RMS $z_s - z_{us}$ (m)	Improvement in RMS $z_s - z_{us}$ (%)
COAGWO-LQR	0.522	0.1041	77.40	0.009686	0.001893	15.26
GWO-LQR	0.777	0.1654	64.09	0.007053	0.001508	32.49
BA-LQR	2.034	0.3176	31.06	0.005796	0.00146	34.64
ACO-LQR	0.8292	0.1525	66.89	0.008322	0.001736	22.29
AIWPSO-LQR	1.991	0.3125	32.16	0.005860	0.001489	33.34
APPO-LQR	1.997	0.3145	31.73	0.00573	0.001435	35.76
Passive	1.964	0.4607	–	0.009445	0.002234	–
Controller	PTP $k_s(z_{us} - z_r)$ (N)	RMS $k_s(z_{us} - z_r)$ (N)	Improvement in RMS $k_s(z_{us} - z_r)$ (%)	PTP z_s (m)	RMS z_s (m)	Improvement in RMS z_s (%)
COAGWO-LQR	2.745	0.5378	19.72	0.004554	0.002181	49.11
GWO-LQR	2.254	0.4575	31.70	0.006314	0.00271	36.77
BA-LQR	2.257	0.4581	31.61	0.005204	0.002011	53.07
ACO-LQR	2.441	0.4826	27.95	0.004897	0.002299	46.36
AIWPSO-LQR	2.278	0.4637	30.78	0.005101	0.001983	53.73
APPO-LQR	2.302	0.4669	30.30	0.005305	0.002046	52.26
Passive	3.243	0.6699	–	0.01523	0.004286	–

to other controllers, including ACO-LQR [7], AIWPSO-LQR [8], GWO-LQR [13], BA-LQR [13] and APPO-LQR [15], which show improvements ranging from 46.73% to 78.72%. Moreover, the COAGWO-LQR controller consistently outperforms other controllers in terms of RMS metrics, with improvements ranging from 15.26% to 77.40%. In comparison, the other controllers exhibit varying levels of performance, with improvements ranging from 27.95% to 64.09% when compared to the COAGWO-LQR controller. In a nutshell, these findings underscore the effectiveness of the COAGWO-LQR controller in reducing vibrations and enhancing ride comfort in vehicle suspension systems.

7. Conclusion and future directions

Researchers have been focused on tackling active suspension control problems to meet challenging control objectives, including body acceleration, suspension deflection, and tire deflection. In recent years, metaheuristic algorithms have been employed to solve the weight optimization problem of LQR applied to address the conflicting control objectives of active vehicle suspension systems. In this paper, the GWO algorithm is combined with COA to allow wolves to have climbing ability, thereby improving exploration in the search space. The exploitation ability of GWO and the exploration ability of COA have been integrated into the proposed hybrid algorithm (COAGWO) to globally find solutions to optimization problems.

The performance of COAGWO has been evaluated by employing CEC-C06 2019 benchmark problems. The experimental outcomes demonstrate that COAGWO achieves quicker convergence and greater accuracy than advanced algorithms such as GWO, COA AGWO, RSA, WOA, TSA, and SOA in the literature. Furthermore, the results from the statistical analysis of Wilcoxon's rank-sum and one-way analysis of variance (ANOVA) tests revealed significant differences among the algorithms, indicating that there are likely meaningful variations in the obtained experimental results.

In addition, the LQR controller is an essential method to improve the handling stability and comfort of active suspension systems. Nevertheless, the selection of weighting matrices Q and R using trial and error may not yield optimal performance in standard LQR control. In this work, the proposed CAOGWO is employed to solve the weighting matrices problem of the LQR controller for active suspension systems. The results indicate that the COAGWO-tuned LQR controller effectively enhances vehicle ride comfort by decreasing the RMS values of the body acceleration by 77.57%, 90.06%, and 77.40% under square, bumpy, and class C-road profiles, respectively, in comparison to the passive suspension. Moreover, validation has been performed by comparing the response of the COAGWO-tuned controller with results from published works using GWO, BA, ACO, AIWPSO, and APPO algorithms, confirming the clear superiority of the proposed COAGWO algorithm. The comprehensive analyses have confirmed the effectiveness of the proposed algorithm in optimizing the LQR controller for the suspension system.

Meta-heuristic algorithms cannot effectively solve every optimization problem. This limitation also applies to the proposed COAGWO algorithm, which indicates the potential necessity for further enhancements, adjustments, and revisions to address various optimization challenges. In future work, it is possible to design binary and multi-objective versions of the proposed COAGWO algorithm for a wide range of real-world problems. Moreover, it is recommended to apply the proposed COAGWO algorithm to challenging control problems such as magnetic levitation, quadcopter, aircraft pitch angle control, and regulating the speed of direct current motors.

CRediT authorship contribution statement

Hasan Başak: Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Methodology, Formal analysis, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors would like to thank the editors and anonymous reviewers for providing insightful suggestions and comments to improve the quality of this research paper.

Appendix. Abbreviation list

Abbreviation	Meaning
ACO	Ant Colony Optimization
AGWO	Aquila-Grey Wolf Optimizer
AIWPSO	Adaptive Inertia Weight Particle Swarm Optimization
APPO	Adaptive Predator–Prey Optimization
BA	Bat Algorithm
COA	Coati Optimization Algorithm
COAGWO	Coati–Grey Wolf Optimization
DF	Degrees of Freedom
F_a	F-statistic
GWO	Grey Wolf Optimization
LQR	Linear Quadratic Regulator
MS	Mean Square
PTP	Peak-to-Peak
RSA	Reptile Search Algorithm
RMS	Root Mean Square
SOA	Seagull Optimization Algorithm
SS	Sum of Squares
TSA	Tunicate Swarm Algorithm
WOA	Whale Optimization Algorithm

References

[1] V.S. Deshpande, B. Mohan, P. Shendge, S. Phadke, Disturbance observer based sliding mode control of active suspension systems, J. Sound Vib. 333 (11) (2014) 2281–2296.

[2] I. Fialho, G.J. Balas, Road adaptive active suspension design using linear parameter-varying gain-scheduling, IEEE Trans. Control Syst. Technol. 10 (1) (2002) 43–54.

[3] X. Shao, F. Naghdy, H. Du, Reliable fuzzy H_∞ control for active suspension of in-wheel motor driven electric vehicles with dynamic damping, Mech. Syst. Signal Process. 87 (2017) 365–383.

[4] J. Cao, P. Li, H. Liu, An interval fuzzy controller for vehicle active suspension systems, IEEE Trans. Intell. Transp. Syst. 11 (4) (2010) 885–895.

[5] M.P. Nagarkar, Y.J. Bhalerao, G.J. Vikhe Patil, R.N. Zaware Patil, GA-based multi-objective optimization of active nonlinear quarter car suspension system—PID and fuzzy logic control, Int. J. Mech. Mater. Eng. 13 (2018) 1–20.

[6] A. Unger, F. Schimmack, B. Lohmann, R. Schwarz, Application of LQ-based semi-active suspension control in a vehicle, Control Eng. Pract. 21 (12) (2013) 1841–1850.

[7] S. Manna, G. Mani, S. Ghildiyal, A.A. Stonier, G. Peter, V. Ganji, S. Murugesan, Ant colony optimization tuned closed-loop optimal control intended for vehicle active suspension system, IEEE Access 10 (2022) 53735–53745.

[8] J.S. David Reddipogu, V.K. Elumalai, Hardware in the loop testing of adaptive inertia weight PSO-tuned LQR applied to vehicle suspension control, J. Control Sci. Eng. 2020 (2020) 1–16.

[9] A.E. Bryson, Control of Spacecraft and Aircraft, vol. 41, Princeton university press Princeton, 1993.

[10] H. Maghfiroh, M. Nizam, M. Anwar, A. Ma'Arif, Improved LQR control using PSO optimization and Kalman filter estimator, IEEE Access 10 (2022) 18330–18337.

[11] K. Hassani, W.-S. Lee, Multi-objective design of state feedback controllers using reinforced quantum-behaved particle swarm optimization, Appl. Soft Comput. 41 (2016) 66–76.

[12] X. Shao, J. Zhang, X. Zhang, et al., Takagi–Sugeno fuzzy modeling and PSO-based robust LQR anti-swing control for overhead crane, Math. Probl. Eng. 2019 (2019) 1–14.

[13] T. Yuvapriya, P. Lakshmi, V.K. Elumalai, Experimental validation of LQR weight optimization using bat algorithm applied to vibration control of vehicle suspension system, IETE J. Res. 69 (2023) 8142–8152.

[14] W. Zhao, L. Gu, M. Dong, Application of artificial fish swarm algorithm in LQR control for active suspension, in: 2022 13th Asian Control Conference, ASCC, IEEE, 2022, pp. 2406–2409.

[15] R.R. Das, V.K. Elumalai, R.G. Subramanian, K.V.A. Kumar, Adaptive predator–prey optimization for tuning of infinite horizon LQR applied to vehicle suspension system, Appl. Soft Comput. 72 (2018) 518–526.

[16] D.H. Wolpert, W.G. Macready, No free lunch theorems for optimization, IEEE Trans. Evol. Comput. 1 (1) (1997) 67–82.

[17] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of ICNN'95-International Conference on Neural Networks, Vol. 4, IEEE, 1995, pp. 1942–1948.

[18] S. Mirjalili, S. Mirjalili, Genetic algorithm, in: Evolutionary Algorithms and Neural Networks: Theory and Applications, Springer, 2019, pp. 43–55.

[19] M. Dorigo, C. Blum, Ant colony optimization theory: A survey, Theor. Comput. Sci. 344 (2–3) (2005) 243–278.
[20] S. Das, A. Biswas, S. Dasgupta, A. Abraham, Bacterial foraging optimization algorithm: theoretical foundations, analysis, and applications, in: Foundations of computational intelligence volume 3: global optimization, Springer, 2009, pp. 23–55.
[21] X.-S. Yang, Firefly algorithm, stochastic test functions and design optimisation, Int. J. Bio-Inspir. Comput. 2 (2) (2010) 78–84.
[22] E. Rashedi, H. Nezamabadi-Pour, S. Saryazdi, GSA: a gravitational search algorithm, Inf. Sci. 179 (13) (2009) 2232–2248.
[23] S. Li, H. Chen, M. Wang, A.A. Heidari, S. Mirjalili, Slime mould algorithm: A new method for stochastic optimization, Future Gener. Comput. Syst. 111 (2020) 300–323.
[24] A.A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, H. Chen, Harris hawks optimization: Algorithm and applications, Future Gener. Comput. Syst. 97 (2019) 849–872.
[25] L. Wang, X.-l. Zheng, A knowledge-guided multi-objective fruit fly optimization algorithm for the multi-skill resource constrained project scheduling problem, Swarm Evol. Comput. 38 (2018) 54–63.
[26] L. Wu, Q. Liu, X. Tian, J. Zhang, W. Xiao, A new improved fruit fly optimization algorithm IAFOA and its application to solve engineering optimization problems, Knowl.-Based Syst. 144 (2018) 153–173.
[27] R. Hu, S. Wen, Z. Zeng, T. Huang, A short-term power load forecasting model based on the generalized regression neural network with decreasing step fruit fly optimization algorithm, Neurocomputing 221 (2017) 24–31.
[28] G. Ding, F. Dong, H. Zou, Fruit fly optimization algorithm based on a hybrid adaptive-cooperative learning and its application in multilevel image thresholding, Appl. Soft Comput. 84 (2019) 105704.
[29] S. Malakar, M. Ghosh, S. Bhowmik, R. Sarkar, M. Nasipuri, A GA based hierarchical feature selection approach for handwritten word recognition, Neural Comput. Appl. 32 (2020) 2533–2552.
[30] R. Sihwail, K. Omar, K.A.Z. Ariffin, M. Tubishat, Improved Harris Hawks optimization using elite opposition-based learning and novel search mechanism for feature selection, IEEE Access 8 (2020) 121127–121145.
[31] R. Joseph Manoj, M. Anto Praveena, K. Vijayakumar, An ACO–ANN based feature selection algorithm for big data, Cluster Comput. 22 (2019) 3953–3960.
[32] X. Zhang, Y. Xu, C. Yu, A.A. Heidari, S. Li, H. Chen, C. Li, Gaussian mutational chaotic fruit fly-built optimization and feature selection, Expert Syst. Appl. 141 (2020) 112976.
[33] X.-l. Zheng, L. Wang, S.-y. Wang, A novel fruit fly optimization algorithm for the semiconductor final testing scheduling problem, Knowl.-Based Syst. 57 (2014) 95–103.
[34] H. Hu, Y. Ao, Y. Bai, R. Cheng, T. Xu, An improved Harris's Hawks optimization for SAR target recognition and stock market index prediction, IEEE Access 8 (2020) 65891–65910.
[35] Y. Wei, H. Lv, M. Chen, M. Wang, A.A. Heidari, H. Chen, C. Li, Predicting entrepreneurial intention of students: An extreme learning machine with Gaussian barebone Harris Hawks optimizer, IEEE Access 8 (2020) 76841–76855.
[36] S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey Wolf optimizer, Adv. Eng. Softw. 69 (2014) 46–61.
[37] N. Hatta, A.M. Zain, R. Sallehuddin, Z. Shayfull, Y. Yusoff, Recent studies on optimisation method of Grey Wolf optimiser (GWO): a review (2014–2017), Artif. Intell. Rev. 52 (2019) 2651–2683.
[38] M.H. Sulaiman, Z. Mustaffa, M.R. Mohamed, O. Aliman, Using the gray wolf optimizer for solving optimal reactive power dispatch problem, Appl. Soft Comput. 32 (2015) 286–292.
[39] Y. Tikhamarine, D. Souag-Gamane, A.N. Ahmed, O. Kisi, A. El-Shafie, Improving artificial intelligence models accuracy for monthly streamflow forecasting using Grey Wolf optimization (GWO) algorithm, J. Hydrol. 582 (2020) 124435.
[40] A.K.M. Khairuzzaman, S. Chaudhury, Multilevel thresholding using Grey Wolf optimizer for image segmentation, Expert Syst. Appl. 86 (2017) 64–76.
[41] P. Dutta, S.K. Nayak, Grey Wolf optimizer based PID controller for speed control of BLDC motor, J. Electr. Eng. Technol. 16 (2021) 955–961.
[42] J. Agarwal, G. Parmar, R. Gupta, A. Sikander, Analysis of Grey Wolf optimizer based fractional order PID controller in speed control of DC motor, Microsyst. Technol. 24 (2018) 4997–5006.
[43] S. Dhargupta, M. Ghosh, S. Mirjalili, R. Sarkar, Selective opposition based Grey Wolf optimization, Expert Syst. Appl. 151 (2020) 113389.
[44] R. Purushothaman, S. Rajagopalan, G. Dhandapani, Hybridizing gray wolf optimization (GWO) with grasshopper optimization algorithm (GOA) for text feature selection and clustering, Appl. Soft Comput. 96 (2020) 106651.
[45] H. Mohammed, T. Rashid, A novel hybrid GWO with WOA for global numerical optimization and solving pressure vessel design, Neural Comput. Appl. 32 (18) (2020) 14701–14718.
[46] C. Ma, H. Huang, Q. Fan, J. Wei, Y. Du, W. Gao, Grey Wolf optimizer based on Aquila exploration method, Expert Syst. Appl. 205 (2022) 117629.
[47] M. Dehghani, Z. Montazeri, E. Trojovská, P. Trojovskỳ, Coati optimization algorithm: A new bio-inspired metaheuristic algorithm for solving optimization problems, Knowl.-Based Syst. 259 (2023) 110011.
[48] F. Yan, J. Xu, K. Yun, Dynamically dimensioned search Grey Wolf optimizer based on positional interaction information, Complexity 2019 (2019) 1–36.
[49] K. Price, N. Awad, M. Ali, P. Suganthan, Problem definitions and evaluation criteria for the 100-digit challenge special session and competition on single objective numerical optimization, in: Technical Report, Nanyang Technological University Singapore, 2018.
[50] L. Abualigah, M. Abd Elaziz, P. Sumari, Z.W. Geem, A.H. Gandomi, Reptile search algorithm (RSA): A nature-inspired meta-heuristic optimizer, Expert Syst. Appl. 191 (2022) 116158.
[51] S. Mirjalili, A. Lewis, The whale optimization algorithm, Adv. Eng. Softw. 95 (2016) 51–67.
[52] S. Kaur, L.K. Awasthi, A. Sangal, G. Dhiman, Tunicate swarm algorithm: A new bio-inspired based metaheuristic paradigm for global optimization, Eng. Appl. Artif. Intell. 90 (2020) 103541.
[53] G. Dhiman, V. Kumar, Seagull optimization algorithm: Theory and its applications for large-scale industrial engineering problems, Knowl.-Based Syst. 165 (2019) 169–196.
[54] J. Apkarian, A. Abdossalami, Active Suspension Experiment for MATLAB/Simulink Users–Laboratory Guide, Quanser Inc., Markham, Ontario, Canada, 2013.
[55] Q. Fu, J. Wu, C. Yu, T. Feng, N. Zhang, J. Zhang, Linear quadratic optimal control with the finite state for suspension system, Machines 11 (2) (2023) 127.