A fuzzy adaptive controller for cuckoo search algorithm in active suspension system

Chun-Tang Chao, Ming-Tang Liu, Chi-Jo Wang and Juing-Shian Chiou

Abstract
This paper presents a fuzzy adaptive cuckoo search algorithm to improve the cuckoo search algorithm, which may easily fall into a local optimum when handling multiobjective optimization problems. The Fuzzy–Proportional-Integral-Derivative (PID) controller design for an active micro-suspension system has been incorporated into the proposed fuzzy adaptive cuckoo search algorithm to improve both driving comfort and road handling. In the past research, a genetic algorithm was often applied in Fuzzy–PID controller design. However, when the dimension is high and there are numerous local optima, the traditional genetic algorithm will not only have a premature convergence, but may also be trapped in the local optima. In the proposed fuzzy adaptive cuckoo search, all parameters of the PID controller and fuzzy rules are real coded to 75 bits in the optimization problem. Moreover, a fuzzy adaptive strategy is proposed for dynamically adjusting the learning parameters in the fuzzy adaptive cuckoo search, and this indeed enables global convergence. Experimental results verify that the proposed fuzzy adaptive cuckoo search algorithm can shorten the computing time in the evolution process and increase accuracy in the multiobjective optimization problem.

Keywords
Cuckoo search, PID controller, particle swarm algorithm, fuzzy, suspension system

Introduction
Engineering design problems often involve highly nonlinear constraints and are related to multiobjective optimization problems. In dealing with these real-world optimization problems, many engineers still set or adjust parameters through past knowledge or experience for single objective optimization. When operating in this manner, compromises and approximations are required, but true optimization is hard to achieve, as these objectives are usually in conflict with each other.

In academia, the area of optimization has received much attention in recent decades. Many various new algorithms have been developed and applied in the industry, which leads to savings on both time and cost. Moreover, many real-world problems are NP-hard, which can be effectively solved by using metaheuristic algorithms based on intelligent knowledge. Currently, evolutionary algorithms, inspired by events in nature, have gained enormous attention in the field of intelligent computing. The genetic algorithm (GA) and particle swarm algorithm (PSO) are traditional well-known algorithms, while the immune algorithm and cuckoo search (CS) are subsequent algorithms. These metaheuristic algorithms play a major role in multiobjective global optimization problems.
The PSO was proposed by Kennedy and Eberhart in 1995, which is an optimization technique based on swarm intelligence. Xin et al. proposed several classification mechanisms and a versatile taxonomy to differentiate and analyze various hybridization strategies on differential evolution and PSO. Chih et al. also proposed a new PSO algorithm for solving economic statistical designs of control charts, which belong to a mixed discrete-continuous nonlinear problem. A penalty objective function was defined to encourage the particles to explore both the feasible and infeasible regions near the border of the feasible area. To further solve the multidimensional knapsack problem (MKP), Chih et al. proposed two PSO algorithms based on a binary PSO with time-varying acceleration coefficients. These two PSO algorithms were used as the foundation to support the novel self-adaptive check and repair operator methods. Therefore, changing different types of pseudo-utility ratios produces solutions with better results when solving MKP. On the other hand, Ciomei and Kyriakides presented a novel heuristic-hybrid optimization method, which combined API (a special class of ant colony optimization) and RCGA (real-coded genetic algorithm), to solve the nonconvex economic dispatch problem in power systems. The downhill behavior of API will avoid the GA search strategy being trapped in local optima. In 2016, Garg proposed an efficient hybrid PSO–GA algorithm for constrained optimization problems with the help of the parameter-free penalty function. In this PSO–GA, PSO works toward improving the vector, while GA has been used for modifying the decision vectors using genetic operators.

Yang and Deb extended the CS for single optimization to the multiobjective cuckoo search (MOCS) for solving multiobjective problems. The superiority of MOCS can be attributed to the fact that the CS uses a combination of vectorized mutation, crossover by permutation and Lévy flights, and selective elitism among the best solutions. Jin et al. developed a CS-based PID controller with investigation on the parameters of the CS including nest number, discovered probability, and step length.

Basically, PSO has the fast convergence characteristic and the drawback of easily getting trapped in the local optimum, which is an especially important issue for multiobjective global optimization problems. In the proposed fuzzy adaptive cuckoo search (FACS) algorithm, the nest positions are real coded to evolve to an optimum. When the FACS is applied to Fuzzy–PID controller design for an active suspension system, all the parameters relating to fuzzy rules or PID controller are real coded to 75 bits. With the incorporation of roulette wheel selection and the proposed fuzzy adaptive strategy, the learning parameters, such as learning rate or step size, will be adaptively decided, and this causes the FACS to effectively increase the searching ability and directions to enable global convergence and optimization accuracy. The rest of the paper is organized as follows: “Fuzzy adaptive cuckoo search algorithm” section presents the proposed FACS algorithm, with the real coding and the fuzzy adaptive strategy included. In “The optimal control problem” section, the optimal control problem and structure of an active suspension system is described and configured. The simulation results are shown in “Simulation results” section. The final section offers brief concluding comments.

**Fuzzy adaptive cuckoo search algorithm**

After a study of the reproductive behavior of cuckoo birds and characteristics of Lévy flight, Yang and Deb developed the CS algorithm at the University of Cambridge in 2009. Preliminary research shows that CS is very promising and outperforms some existing algorithms, such as GA and PSO.

Some species of cuckoos are brood parasites, and they neither build their own nests nor hatch their eggs, but specialize and lay eggs that closely resemble the eggs of their chosen host. Some host birds are able to distinguish cuckoo eggs from their own, and they will either throw those alien eggs away or simply abandon their nests and build a new nest elsewhere, which will influence the egg’s survival rate and the cuckoo population in the next generation.

In nature, the search for food by animals or insects is in a random or quasi-random manner. This is similar to the flight behavior of birds, where flight direction depends implicitly on a random probability. Lévy flight is a specialized behavioral theory that looks at random animal or insect flight and is applied in the mathematical model of the CS algorithm for simulating the reproductive behavior of cuckoo birds. In general, Lévy flights are a random walk and Lévy step is drawn from the Lévy distribution. During the flight, a larger step size will expand the scope of search and enable diversity of the population, which is suitable for the early stages of the search to avoid falling into a local optimum. In the later period of search, a smaller step size is appropriate for converging to the global optimal solution. The proposed FACS algorithm is outlined as follows:

**Step 1.** Suppose cuckoos lay \( n \) eggs at a time and dump these eggs in randomly chosen \( m \) host nests. The parameters for a Fuzzy–PID controller, such as fuzzy membership functions (left triangle, right triangle, triangle,
Gaussian, sigmoid, reverse sigmoid), fuzzy linguistic variables (NL, NM, NS, ZE, PS, PM, and PL), PID parameters \((k_p,k_i,k_d)\), and the input universe for fuzzy membership functions \((\mu_{e(i)}: U_{e(i)} \rightarrow \{0,1\})\) and \(\mu_{p(i)}: U_{p(i)} \rightarrow \{0,1\}\) are formulated as the position parameters of host nests. The initial population is randomly generated, and the real code is applied with each bit representing a real number. Each cuckoo lays one egg in a nest, and each egg relates to the target solution with 75 parameters (bits) in all. The fitness values for \(m\) host nests are evaluated, and the best fit nest will be kept in the next generation. The iterative algorithm is proceeded to calculate the new position of cuckoo nests and then find the global optimal solution.

**Step 2.** In nature, it is random for cuckoos to look for suitable nests to lay their eggs. In order to simulate this parasitic habit, the following three idealized rules are used:

a. A cuckoo only lays one egg at a time and randomly finds a nest to hatch it.

b. The egg in the best nest will be kept in the next generation.

c. For \(m\) host nests, the host birds will find those alien eggs with a probability \(p_a \in [0,1]\). In this case, the host birds may either throw those alien eggs away or abandon the nests and build new ones in new locations.

Based on the above three idealized rules, the new search space \(X_i(t + 1)\) at the time \(t\) will be calculated as

\[
X_i(t + 1) = X_i(t) + \kappa \otimes \text{Levy}(\lambda), \quad i = 1, 2, \ldots, n,
\]

where \(\kappa > 0\) helps in deciding the step size. The random walk \(\text{Levy}(\lambda)\) drawn from Lévy flight is a concept from a Markov chain, where the next location \(X_i(t + 1)\) only depends on the current location \(X_i(t)\). Generally, the Lévy step is taken from the Lévy distribution, which is expressed in power form by Yang, as shown in equation (2)

\[
\text{Levy}(\lambda) \sim \mu = t^{-\lambda}, \quad (1 < \lambda \leq 3)
\]

A simple scheme to obtain the Lévy step is from the Mantegna algorithm in 1992

\[
a = \frac{\mu}{|\nu|^{1/\beta}},
\]

where \(a\) is Lévy(\(\lambda\)) in equation (1) and \(\beta = \lambda - 1, \beta \in (0, 2]\). Moreover, the random variables \(u\) and \(v\) are of normal distributions, \(u \sim N(0, \sigma_u^2)\) and \(v \sim N(0, \sigma_v^2)\), with standard deviations shown below

\[
\sigma_u = \left[ \frac{\Gamma(1+\beta) \times \sin(\beta \pi/2)}{\Gamma((1+\beta)/2) \times 2^{\beta/2} \times \beta \pi} \right]^{1/\beta}, \quad 0 < \beta \leq 2
\]

\[
\sigma_v = 1
\]

**Step 3.** Evaluation of the Lévy random flight mechanism in equations (2) to (4) gives leads to new search space for the FACS algorithm in equation (1). The worst nests are abandoned by host birds with a probability \(p_a\) and new ones are built. The step size \(\kappa \otimes \text{Levy}(\lambda) = \kappa \otimes a\) and discovered probability \(p_a\) will be evaluated and updated for the next generation.

**Step 4.** Elitism is performed by keeping the best fit result in each iteration. The roulette wheel selection technique is used to determine the new position of the cuckoo nests for the next generation.

**Step 5.** Go to Step 2 and repeat the process until the termination criterion of the FACS algorithm is satisfied.

**The coding of FACS**

In this paper, the real coding is applied for real-parameter processing in FACS, instead of binary coding. There are 75 real parameters, representing fuzzy memberships and fuzzy linguistic variables, in each target solution for the Fuzzy–PID controller design.

The coding bits from bit 1 to bit 75 are denoted as \(S_1(N), \ldots, S_{21}(N), S_{22}(N), \ldots, S_{70}(N), S_{71}(N), S_{72}(N), S_{73}(N), S_{74}(N), S_{75}(N)\). Fuzzy membership functions can be described by \(S_1(N) \sim S_{21}(N)\), where every three bits represent a fuzzy membership function. For example, \(S_1(N)\) represents the “start position” and
$S_2(N)$ is on behalf of the “stop position;” thus, the center point is defined as follows

$$C(N) = S_1(N) + \frac{S_2(N) - S_1(N)}{2}$$ (5)

$S_3(N)$ stands for “function shape” with an integer from 0 to 6, such as “1”: triangle, “2”: left triangle, “3”: right triangle, “4”: Gaussian, “5”: sigmoid, “6”: reverse sigmoid, and “0”: invalid, where the last ‘invalid shape” is to avoid unnecessary calculation and fuzzy rules.

As for fuzzy linguistic variables, such as NL, NM, NS, ZE, PS, PM, and PL, they are coded by an integer from 1 to 7. All fuzzy rules will be described by 49 bits, i.e. $S_{22}(N) \sim S_{70}(N)$. Moreover, the PID parameters $(k_p, k_i, k_d)$ and the input universe $(k_e(t), k_{de}(t))$ are represented by $S_{71}(N) \sim S_{75}(N)$. In this paper, it was set to evolve into 20 groups in each iteration, and each group has its respective target solution with 75 bits defining the above real parameters. In fact, the group number is exactly the number of host nests $m$. Figure 1 shows the flow chart of the proposed FACS.

### The fuzzy adaptive strategy

For the discovered probability $p_a$ and the parameter $\beta$, they will both affect the search step size, and it will be unfavorable to the local and global search strategies of the CS algorithm if both of them are set to be fixed values. To make the CS algorithm more efficient in search performance, the proposed FACS will adjust $p_a$ and $\beta$ in a dynamic manner. In the early period of the FACS, $p_a$ is set to be smaller to make relatively more nests updated, and this will also maintain a stronger global search capability for the algorithm. Simultaneously, $\beta$ is also set to be
smaller to increase the global search power with larger step size. Although in this case, there will be less accuracy in local search and poorer convergence. However, it is beneficial to fast evolution in the early period of FACS. On the contrary, in the late period of the FACS, larger \( p_u \) can cause relatively fewer nests to be updated and maintain a stronger local search capability for the algorithm. Also, larger \( \beta \) results in smaller step size and increases accuracy in the local search. Based on the above discussion, we proposed fuzzy rules for tuning \( p_u \) and \( \beta \). Table 1 shows the fuzzy rule table, where \( BF \) represents the number of times that the best-fit value appears in each generation, \( VF \) relates the number of changes for fitness value, and \( UN \) is the number of non-changing fitness values after several generations. For fuzzy sets \( BF, VF, \) and \( UN \), each is described by three triangle membership functions: L (Low), M (Medium), and H (High). The intervals for these linguistic variables or parameter are \( BF: [0,4] \), \( VF: [0,5] \), \( UN: [0,10] \), \( p_u: [0,1] \), and \( \beta: [0,2] \). Finally, equation (6) shows one fuzzy rule example, which is highlighted in red in Table 1.

If \( UN \) is High and \( VF \) Medium, then \( Pa \) is Low and \( \beta \) is Medium

**The optimal control problem**

Automotive electronics have attracted tremendous attention and investment in the last few years, especially the active suspension system. At present, some traditional and passive suspension systems, with springs and dampers, have been installed or totally replaced by a force actuator with advanced control methodologies. The active suspension system possesses the ability to improve handling stability and driving comfort. However, there usually exists nonlinear and uncertain dynamic characteristics in the suspension system. With an increase in the damping force of the shock absorber, automobile stability will improve and reduce road shock and vibration. But when we desire to offer comfort for passengers on the straight and smooth pavement, the damping force of the shock absorber should be smaller. So the trade-off between comfort and handling is an important issue to the suspension system. To verify the performance and feasibility of the proposed FACS, a Fuzzy–PID controller was designed to solve the multiobjective optimization problem. Figure 2 shows the active micro-suspension system model, which has been widely used in micro-suspension system design of the quarter-car model\(^{12-14}\).

The dynamic equation of the active suspension system is

\[
m_s \ddot{z}_s = k_s (z_u - z_s) + b_s (\dot{z}_u - \dot{z}_s) + f_u
\]

\[
m_u \ddot{z}_u = -k_s (z_u - z_s) - b_s (\dot{z}_u - \dot{z}_s) - f_u + k_i (z_r - z_u)
\]

where

- \( z_r \) = road disturbance input,
- \( z_s \) = sprung mass displacement,
- \( z_u \) = unsprung mass displacement, and
- \( f_u \) = controller output damping force, and the parameter setting is shown in Table 2.\(^{12}\)

To simulate a realistic road surface, pseudorandom road profiles are simulated using stochastic processes (PSD, power spectral density function), as shown in equation (9).

\[
S(m) = \frac{CP \sin(\omega_m t)}{(\omega_m)^{3.5}}
\]
where $m$ is the wave number (cycle/m), $w_c \approx 7.7$ (rad/s) is the sprung mass natural frequency, $V$ (m/s) denotes the vehicle speed, $t_s = 0.01$ s is the sampling time of simulation, $i$ is the iteration step, $P = 3.14 \times 10^{-6}$ is the road quality, and $C = 10$ is the current road disturbance coefficient. With the assumption that the tire will not leave the ground, the sprung mass displacement $z_s$ and unsprung mass displacement $z_u$ can be measured and controlled to track the desired reference $R = 0$, where $R$ is the reference target input. The road profile shown in Figure 3 with road bump $z_r$ and road roughness $\dot{z}_r$ can simulate the driver’s actions on bumpy roads according to different conditions and test how the active suspension system we designed performs superior performance. Several important time periods are listed and described below.

1. During $t = 0.01–1.15$ s, $V$ is 25 (m/s). Then the car begins to accelerate, and $V$ becomes 30 (m/s) during $t = 1.16–2.01$ s.
2. During $t = 2.02–2.87$ s, the extremely bad roads are simulated by PSD.
3. During $t = 2.88–4.02$ s, $V$ is 35 (m/s). During $t = 4.03–5.00$ s, $V$ is 40 (m/s).
4. During $t = 5.01–5.86$ s, the extremely bad roads are simulated by PSD.
5. During $t = 5.87–7.00$ s, $V$ is 30 (m/s). Then the car begins to decelerate, and $V$ becomes 25 (m/s) during $t = 7.01–8.00$ s.
For an active suspension system, the road profile is the input and the outputs are $z_s$ and $z_u$, which will determine handling stability and driving comfort. When the suspension system has an upward shift, the acceleration of the spring mass should be reduced, and vice versa. On the other hand, the automobile handling can be indicated by the tire deflection, which reflects the amount of friction and the degree of exposure between the tire and the ground. Based on the above principles, the fitness function $F$ is defined as follows

$$F = \sum_{n=0}^{800} \left( e[n]^2 + \dot{e}[n]^2 \right)$$  (10)

The simulation time for the well-known step-like and pseudorandom road profile in equation (9) is set as 8 s, thus we have “800” samples in equation (10) with $t_s = 0.01$. Moreover, the system error and error derivative can be evaluated through equations (11) to (13)

$$y_1[n] = z_u[n] - z_r[n], \quad y_2[n] = z_s[n] - z_u[n]$$  (11)

$$e_1[n] = R - y_1[n], \quad e_2[n] = R - y_2[n]$$  (12)

$$e[n] = e_1[n] - e_2[n], \quad \dot{e}[n] = \dot{e}_1[n] - \dot{e}_2[n]$$  (13)

Finally, another important issue is to make sure that there is enough space between the spring mass and the tires, so it can prevent the tires from hitting the car body. A penalty strategy is set up in the system, limiting $(z_s[n] - z_u[n]) \leq 5$ cm.

**Simulation results**

In this paper, the Fuzzy–PID controller design for an active micro-suspension system has been incorporated into the proposed FACS algorithm. Figure 4 shows the proposed Fuzzy–PID control structure, where the optimal real parameters $[S_1, S_2, S_3, \ldots, S_{70}, S_{71}, \ldots, S_{75}]$ will be obtained by the proposed FACS. In the simulation, the number of host nests $m$ is set to 20.

The pavement responses with passive control, the proposed FACS control, and the conventional optimal linear feedback control are shown in Figure 5, including suspension deflection, sprung mass acceleration, and tire-ground beating distance. After evolution of 24 generations, we obtained the final optimal parameters $[S_1, S_2, S_3, \ldots, S_{70}, S_{71}, \ldots, S_{75}] = [-8, -7, 5, -10, -4, 6, -10, 3, 1, -4, 4, 0, -5, 8, 3, -1, 8, 6, 9, 9, 5, 5, 4, 6, 6, 2, 2, 7, 7, 4, 6, 6, 7, 6, 4, 2, 5, 6, 6, 2, 3, 3, 6, 1, 7, 4, 6, 2, 4, 4, 3, 3, 1, 2, 7, 6, 1, 3, 1, 1, 4, 4, 5, 4, 3, 2, 6, 3, 2, 7, -36.1206, 712.8864, 969.7563, 1.0860, 0.8781], where the first 70 bits relates to fuzzy rules, and the last 5 bits corresponds to $(k_p, k_i, k_d, k_{e(t)}, k_{de(t)}) = [-36.1206, 712.8864, 969.7563, 1.0860, 0.8781]$. From Figure 5, it can be first observed that the maximum suspension deflection 0.0391 m, within limit $(z_s[n] - z_u[n]) \leq 5$ cm, has been improved by 21.49% with the proposed FACS when compared with the uncontrolled situation. Subsequently, the maximum suspension acceleration 2.411 m/s$^2$ shows that the comfort of
passengers has been improved by 50.26% with the proposed FACS when compared with no control. Finally, in comparison with the absence of control, the maximum tire-ground contact distance indicates that the automobile handling has been improved by 26.08% with the proposed FACS. Figure 6 shows the final membership functions decoded from the first 21 bits: 

\[ S_1(N) \sim S_{21}(N): [-8, -7, 5, -10, -4, 6, -10, 3, 1, -4, 4, 0, -5, 8, 3, -1, 8, 6, 9, 9, 5] \]

which corresponds to the design result in a fuzzy system:

\[
\begin{align*}
S_1 & (\text{start position : } -0.8), & S_2 & (\text{stop position : } -0.7), & S_3 & (\text{function shape : } 5 \rightarrow \text{sigmoid}), & S_4 & (-1), & S_5 & (-0.4), \\
S_6 & (6 \rightarrow \text{reverse sigmoid}), & \ldots, & S_{19} & (0.9), & S_{20} & (0.9), & S_{21} & (5 \rightarrow \text{sigmoid})
\end{align*}
\]
The success of the proposed fuzzy adaptive strategy is exhibited in Figure 7. With the initial settings \( p_a = 0.25 \) and \( \beta = 1.5 \), the designed fuzzy rules for dynamically adjusting \( p_a \) and \( \beta \) will avoid falling into a local optimal solution. As shown in Figure 7, through the observation of the change of fitness function in each generation, we can find that the evolution process was trapped in the local convergence in the fourth generation and evolve to a global optimal solution in the 22nd generation. It shows that our designed active suspension system can quickly improve steering stability and driving comfort on extremely bad roads. On the other hand, the PID parameters are limited to \( k_p, k_i, k_d \in [-10000, 10000] \). Once the PID parameters are out of the range, they will be reset to the initial values to avoid creating too large of a search space and difficulty in convergence. In this simulation, the termination criterion is determined by the fitness value. When \( F < 380 \), the system will stop the evolution. Simulation shows that the solution was easily trapped in the local optimum when the fitness value ranged between 400 and 450.

The proposed FACS exhibits its efficiency in escaping from the local optimum and achieving \( F = 373.0060 \) \((1/F = 2.67 \times 10^{-3})\) in the 24th generation. In the Matlab environment, it takes 2736 s for the FACS to finish the optimization evolution of 24 generations, with each generation about 114 s. In comparison with the proposed FACS, our previous GA-based Fuzzy–PID control in 2009\(^{12}\) had a similar performance but was inefficient because it needed 100 generations, 48 h of computing time, and was easily trapped in the local optimum.

Like other bio-inspired algorithms, GA or PSO, the CS must take many generations of evolutionary computing to ensure that the algorithm evolves to an optimum. For the proposed FACS, the roulette wheel selection, popular in GA, is employed to make better results inherited to the next generation, which is crucial in multiobjective optimization problems. This will improve the adaptability of the CS and enable the CS to have the ability to solve complex problems.

For the proposed FACS, the search space of nests is further divided into an inner-boundary and outer-boundary, where the border will be determined by the fitness value \( F_0 = 6000 \). When there is a nest solution with fitness value greater than \( F_0 \), it will be defined as an outer-boundary nest. At this moment, the PID parameters will be reset to the initial values and the system will find the new target solution. This strategy is also to avoid creating too large of a search space and difficulty in convergence. Figure 8 shows the trajectory of PID parameter changes in 24 generations. The final distribution of \( e(t) \) and \( \dot{e}(t) \) of 20 groups is shown in Figure 9.

To address CS shortcomings of low accuracy and poor convergence, especially for multiobjective optimization problems, this paper proposes a novel FACS algorithm, which combines roulette wheel selection and dynamic parameter adjustment to effectively improve accuracy and convergence speed. A fuzzy adaptive strategy is presented for dynamically tuning \( p_a \) and \( \beta \), instead of using fixed values in a traditional CS algorithm, to enable the FACS to prevent from falling into a local optimum, but rather evolve to a global optimal solution. Furthermore, to avoid having too large of a search space and difficulty in global convergence, the PID parameters are limited to \( k_p, k_i, k_d \in [-10000, 10000] \) and the search space is restricted to the nests with fitness value less than \( F_0 = 6000 \). With larger \( F_0 \) and a larger range of values for the PID parameters, it indicates the larger nest search.
space for the cuckoos and results in more generations of evolution, which is unfavorable to fast convergences of a system, but helps in jumping off the local optimum.

**Conclusions**

To solve the problem of nonlinear and uncertain dynamic characteristics in a suspension system, this paper might be the first one to propose a novel CS Algorithm to design a Fuzzy-PID controller for a quarter-car model with suspension system problems simplified to a multiobjective optimization problem. For a CS algorithm, the choice of the nest position is regarded as an important disturbance factor to increase accuracy in the optimization problem. The proposed FACS algorithm yields quicker convergence and enables global convergence for a
practical multiobjective system. The active suspension system we designed has the ability to improve handling stability and driving comfort. The experimental results show that the proposed FACS algorithm has a good application prospect.

In recent years, deep learning approaches have gained significant interests and remarkable advances. The proposed FACS algorithm, a novel meta-heuristic algorithm, may further combine with deep learning networks to construct a more effective and predictive model for active suspension systems.

Author contributions
Ming-Tang Liu and Chun-Tang Chao developed the methodology and drafted the manuscript. Moreover, Chi-Jo Wang and Juing-Shian Chiou derived the formulas and completed the simulations. All the authors approved the final manuscript.

Declaration of conflicting interests
The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding
The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by the Ministry of Science and Technology, Taiwan, under grant nos 106-2221-E-218-001-MY2.

References
1. Eberhart R and Kennedy J. A new optimizer using particle swarm theory. In: Proceedings of the sixth international symposium on micro machine and human science, Nagoya, Japan, 1995, pp.39–43.
2. Kennedy J and Eberhart RC. Particle swarm optimization. In: Proceedings of the IEEE International Conference on Neural Networks, Perth, Australia, IEEE Service Center, Piscataway, NJ, 1995, pp.1942–1948.
3. Xin B, Chen J, Zhang J, et al. Hybridizing differential evolution and particle swarm optimization to design powerful optimizers: a review and taxonomy. IEEE Trans Syst Man Cybern C 2012; 42: 744–767.
4. Chih M, Yeh LL and Li FC. Particle swarm optimization for the economic and economic statistical designs of the X control chart. Appl Soft Comput 2011; 11: 5053–5067.
5. Chih M, Lin CJ, Chern MS, et al. Particle swarm optimization with time-varying acceleration coefficients for the multidimensional knapsack problem. Appl Math Model 2014; 38: 1338–1350.
6. Chih M. Self-adaptive check and repair operator-based particle swarm optimization for the multidimensional knapsack problem. Appl Soft Comput 2015; 26: 378–389.
7. Ciornei I and Kyriakides E. A GA-API solution for the economic dispatch of generation in power system operation. IEEE Trans Power Syst 2012; 27: 233–242.
8. Garg H. A hybrid PSO-GA algorithm for constrained optimization problems. Appl Math Comput 2016; 274: 292–305.
9. Yang XS and Deb S. Multiobjective cuckoo search for design optimization. Comput Oper Res 2013; 40: 1616–1624.
10. Jin Q, Qi L, Jiang B, et al. Novel improved cuckoo search for PID controller design. Trans Inst Measur Control 2015; 37: 721–731.
11. Yang XS and Deb S. Cuckoo search via Lévy flight. In: Proceedings of World congress on nature & biologically inspired computing (NaBIC), India, 2009, pp.210–214.
12. Chiou JS and Liu MT. Using fuzzy logic controller and evolutionary genetic algorithm for automotive active suspension system. Int J Automot Technol 2009; 10: 703–710.
13. Chiou JS and Liu MT. Numerical simulation for fuzzy-PID controllers and helping EP reproduction with PSO hybrid algorithm. Simul Model Pract Theory 2009; 17: 1555–1565.
14. Chiou JS, Tsa SH and Liu MT. A PSO-based adaptive fuzzy PID-controllers. Simul Model Practice Theory 2012; 26: 49–59.
15. Yang XS. Nature-inspired metaheuristic algorithms. Frome: Luniver Press, 2010.
16. Mantegna RN. Fast, accurate algorithm for numerical simulation of Lévy stable stochastic processes. Phys Rev E 1992; 49: 451–458.